ISBN 978-82-326-1687-9 (printed version)
ISBN 978-82-326-1686-2 (electronic version)
ISSN 1503-8181




Doctoral theses at NTNU, 2016:174

Åsmund Ervik

# Multiscale modelling using molecular dynamics and interface-capturing methods for two-phase flow simulation of droplets covered with surfactants or asphaltenes, and applications to electrocoalescence

NTNU
Norwegian University of
Science and Technology
Faculty of Engineering Science and Technology
Department of Energy and Process Engineering

NTNU
Norwegian University of
Science and Technology

Doctoral theses at NTNU, 2016:174

NTNU

NTNU
Norwegian University of
Science and Technology

Åsmund Ervik

# Multiscale modelling using molecular dynamics and interface-capturing methods for two-phase flow simulation of droplets covered with surfactants or asphaltenes, and applications to electro-coalescence

Thesis for the degree of Philosophiae Doctor

Trondheim, June 2016

Norwegian University of Science and Technology
Faculty of Engineering Science and Technology
Department of Energy and Process Engineering

NTNU
Norwegian University of
Science and Technology

**NTNU**
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Engineering Science and Technology
Department of Energy and Process Engineering



ISBN 978-82-326-1687-9 (printed version)
ISBN 978-82-326-1686-2 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2016:174

Printed by Skipnes Kommunikasjon as

## Abstract

THE WORK PRESENTED in this thesis attempts to improve our fundamental understanding of the interfacial properties of water drops in crude-oil. Model fluids consisting of aliphatic oil with surfactants, as well as real crude components such as asphaltenes, are considered. The case of a single drop being deformed, either by an electric field or by a needle deflating the drop, is studied in detail using simulations. Close comparisons with experiments are made, in order to increase confidence in the simulations as a faithful representation of the physical world. Combining methods at different scales, namely the molecular and the continuum, a simple but effective multi-scale method is developed to predict and explain the behaviour of drops with complex interfaces, e.g. water drops in oil covered with asphaltenes.

The first part of this study concerns systems with a deforming drop of water in a model oil with surfactant, where one has a more-or-less complete understanding of the physics and chemistry. These systems are studied in detail using computational fluid dynamics simulations, and direct comparison with experiments is performed. The effect of surfactants on the transient deformation of drops, and in particular on the damping of oscillations, is studied here for the first time. To enable future studies of larger systems with several drops interacting, a 3D parallel version of the code has been developed using a domain decomposition approach. The case of a single falling drop in the presence of surfactants is also considered with a theoretical approach, and exact solutions are obtained for this system.

The second part is the multiscale simulations of systems with a water drop in crude oil. The developed multiscale method consists of coarse-grained molecular dynamics simulations, using the SAFT-$\gamma$ Mie approach, which are used to provide interfacial properties for the continuum simulations. At the continuum scale, a novel hybrid level-set/ghost-fluid/immersed-boundary method has been developed for the simulation of complex fluid-fluid interfaces. Together, these approaches enable a direct link between the chemical composition of the crude oil and the interfacial properties. The molecular structure of crude oil components, and its effect on interfacial properties, is still under debate in the literature. The simulation approach developed here will enable detailed hypothesis testing, which may help settle the debate. A configuration of particular interest in this context is the “crumpling drop”, a phenomenon observed when a water drop in crude oil is drained by means of a pipette. This system is simulated, and good agreement is found with experiments reported in the literature.

For the coarse-grained molecular dynamics approach used in the multiscale approach, namely the SAFT-$\gamma$ Mie force field, two computational tools have been developed. They are called Bottled SAFT and raaSAFT. Together they provide an unprecedented ease-of-use for obtaining models for molecular simulation and for setting up and running these simulations.

## Preface

I HAVE MANY TO THANK for their involvement in this thesis, and fortunately no constraints on the length. The present work forms part of the research project "Fundamental understanding of electrocoalescence in heavy crude oils", informally known as "Electrocoalescence III", coordinated by SINTEF Energy Research. The project is funded by the Research Council of Norway (grant no. 206976) and by the industry partners Statoil, Petrobras and Wärtsilä Oil & Gas Systems. I am very grateful for this support.

I would like to thank my supervisor, Professor Bernhard Müller, for having faith in my ideas and approaches, and for striking a good balance between giving me freedom and advising me to keep my focus whenever I strayed too far off course. I thank also my co-supervisor, Dr. Svend Tollak Munkejord, for his expertise and experience, excellent advice, grammatical corrections, and dry wit.

I am very grateful to the entire project team at SINTEF Energy Research for interesting discussions in our meetings. In particular I want to thank project leader Dr. Marit-Helen Ese, for finding room for a generous travel grant that enabled my stay at Imperial College London. Thanks go also to Dr. Svein Magne Hellesø, whose laboratory and office door has always been open to me. I also consider myself privileged to have been in a project with such interested, knowledgeable and straight-talking industry partner representatives, in particular Dr. Erik Bjørklund at Wärtsilä Oil & Gas Systems. I also want to thank also our collaborators at G2Elab, Dr. Pierre Atten and Professor Jean-Luc Reboud, for providing very useful feedback on our work and for hosting me in Grenoble; thanks also to my fellow PhD student in Grenoble, You Xia, for showing me the laboratory and the city.

My thesis would not have been the same without my six month visit to Imperial College London. I wholeheartedly thank Professor Erich A. Müller at the Department of Chemical Engineering for accepting my proposal for collaboration and for providing generously of his time, as well as the time of his busy postdoctoral researchers. Many thanks go to the entire Molecular Systems Engineering group for truly welcoming me and my family into their midst. In particular, thanks to Dr. Carmelo Herdes, Dr. Maria Guadalupe "Lupe" Jiménez-Serratos and Dr. Thomas F. Headen for teaching me molecular dynamics and for introducing me to the world of asphaltenes. Thanks also to Professor George Jackson, Professor Edo Boek, Professor Omar K. Matar, Dr. Andrew Haslam and Dr. Edward R. Smith for listening to my ideas and for providing useful ideas, suggestions and constructive criticisms of my approach.

Another important contribution came from Morten Olsen Lysgaard, who took on my proposal to do his MSc thesis on the hybrid level-set/ghost-fluid/immersed-boundary method. Taking on a student to supervise was a gamble for me, and thanks to you it paid dividends beyond my expectations. Thank you also for the collaboration since you finished, you keep going above and beyond. I hope the future will bring you back to academia.

I also want to thank my colleagues at the Department of Energy and Process Engineering, in particular Dr. Karl Yngve Lervåg and Dr. Halvor Lund, for help and hints when I started on this journey, and for many enjoyable conversations since. Thanks also to Dr. Dr. Simen Ellingsen, for many a good luncheon, and for the cricket matches. And I must thank my office mate, fellow PhD student Muhammad Rahman, for the interesting discussions and the even more interesting food. In the wider scientific community, I want to thank, among others, Dr. Barry Smith, Dr. Jed Brown and Dr. Matthew Knepley of the PETSc team, and Dr. Joshua Anderson of the HOOMD-blue team; for being the giants on whose shoulders people like myself can stand, and for taking the time to answer all my questions in detail.

A computational thesis is nothing without computing resources. Many simulations have been run on my desktop computers; for this I must thank again my office mate Rahman for not complaining about the noise and heat produced. Other simulations have been run at various high performance computing facilities, and I wish to thank SINTEF Energy Research for computing time on vsl176; NOTUR and NTNU for computing time on Kongull and on Vilje; the Imperial College High Performance Computing Service for computing time on CX1; and NOTUR and the University of Oslo for computing time on Abel. I also want to thank the staff at these locations for their excellent support.

Last, but decidedly not least, I want to thank my family for their support, interest and encouragement during this time. To my brother and sister, Daniel and Åse: you are both sources of inspiration to me, in your pragmatism and your persistence, each in your own way. To my parents, thank you for piquing my interest in science and in computers from an early age, and for always being there. To my wife Elisa, thank you for your unwavering love and support, for being the world's best mother, and for putting your own career on hold when mine took us off to London. To my oldest daughter Aurora, thank you for your questions, your laughs, and all the walks. I've had many ideas sitting on the side of your bed, singing lullabys, waiting for you to fall asleep. And to my youngest daughter Runa, thank you for embodying the term "a bundle of joy".

Trondheim, March 2016
Åsmund Ervik

# Contents


**1 Introduction** 1
- 1.1 A brief history of the electrocoalescer . . . 1
- 1.2 Scientific investigations of electrocoalescence . . . 4
- 1.3 Contributions of the present thesis . . . 7
- 1.4 Applications to other areas . . . 8

**2 Background** 11
- 2.1 The molecular origin of fluid dynamics . . . 11
- 2.2 Incompressible fluid flow . . . 18
- 2.3 Two-phase flow . . . 20
- 2.4 The effect of surfactants on two-phase flow . . . 24
- 2.5 The asphaltenes . . . 27

**3 Computational methods** 31
- 3.1 Methods for incompressible flow . . . 31
- 3.2 Methods for two-phase flow . . . 42
- 3.3 Methods for elastic interfaces . . . 47
- 3.4 Methods for molecular simulation . . . 52

**4 Computational tools** 63
- 4.1 Reproducibility and open science . . . 63
- 4.2 Tools developed in the present work . . . 64
- 4.3 Tools used in the present work . . . 66

**5 Summaries of thesis papers** 71
- 5.1 Paper 1 . . . 71
- 5.2 Paper 2 . . . 75
- 5.3 Paper 3 . . . 77
- 5.4 Paper 4 . . . 79
- 5.5 Paper 5 . . . 83
- 5.6 Paper 6 . . . 87
- 5.7 Paper 7 . . . 91
- 5.8 Paper 8 . . . 93

**6 Conclusions** 97
- 6.1 Conclusions . . . 97
- 6.2 Future outlook . . . 98

**Bibliography** 105

**Appendix** 127
- Paper 1 . . . 129
- Paper 2 . . . 151
- Paper 3 . . . 161
- Paper 4 . . . 169
- Paper 5 . . . 217
- Paper 6 . . . 239
- Paper 7 . . . 271
- Paper 8 . . . 293

CHAPTER 1

# Introduction

## §1.1 A brief history of the electrocoalescer

THE ELECTROCOALESCER is a device used on- or offshore to separate water from crude oil. When oil is extracted from a reservoir, it usually comes together with water, either naturally occurring or injected in order to increase reservoir pressure. When this mixture flows to the surface, it passes reservoir formations and/or pressure reduction valves where the flow is highly turbulent. This flow forms an emulsion, much like the flow around a whisk may form an emulsion of molten chocolate and hot cream in our kitchen. The oil-water emulsion consists of water drops, typically ∼10 micrometres in diameter, dispersed throughout the oil. These drops fall very slowly through the viscous oil, and their interfaces may be stiffened by some components of the oil, for instance molecules called asphaltenes. These effects make it hard to remove the water from the oil.

To speed up the separation, several techniques are available. One is to apply an electric field, as in the electrocoalescer, seen in Figure 1.1.

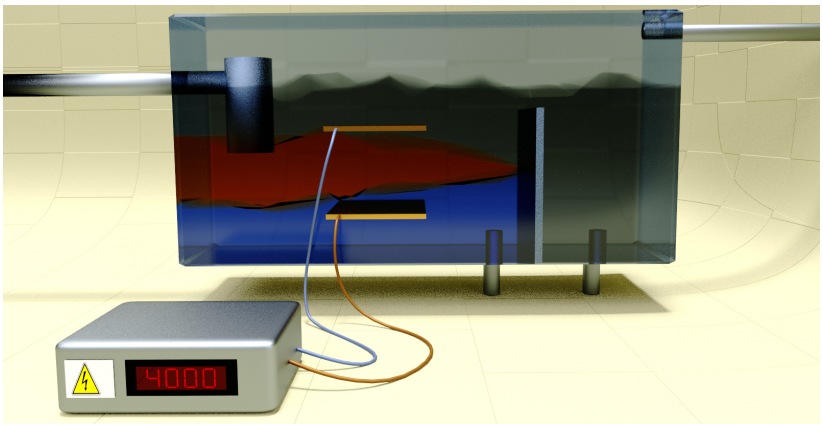



FIGURE 1.1: The author's impression of a three-phase separator with an inline electrocoalescer. Fluid flows in from the left; gas rises and goes out to the right. Oil and water separate partially, with a brown emulsion layer. The applied electric field enhances separation of this layer. Oil and water are held apart by a plate, and go out through pipes at the bottom.

Other techniques include heat-treating the emulsion, adding chemicals to it, centrifugal devices etc. Applying an electric field induces a dipole in all the tiny emulsion drops, which causes an attractive force between them. As the drops coalesce, they become bigger and fall much faster to the bottom where they separate into a clear water layer. But this is not the only working mechanism of the electrocoalescer; it also accelerates the very slow film drainage that may be blocking coalescence of even very close drops.

The history of the electrocoalescer, as well as the history of the chemical additives used to enhance separation, begins around 1905. At the time, the oil boom had spread across the United States. Most oil was very easy to produce, giving us the stereotypical image of the oil well gushing a wide black stream into the air. But some oil was problematic even then. Nicknamed "roily oil", it was a brown sludge that left hopeful oil producers disappointed. One quickly discovered that the product was a mixture of oil and water, so a strategy was devised to recover the oil within: the emulsion was pumped into big trenches and left to sit there. Producers waited months for the mixture to separate, but alas, nothing of the sort was to occur. In the end, the emulsion was simply led out into nearby creeks and discharged into the environment, or it was burned on-site. A photograph of a pool of oil, believed to be an emulsion, is shown in Figure 1.2. This photo is from Glenn Pool oilfield, which was among the more problematic wells in the US.

Emulsions produced in the ocean after oil spills have earned the nickname "chocolate mousse" from clean up workers.

The revolution came in two forms, the electrocoalescer and the chemical demulsifiers. On the west coast of the US, Dr. James B.

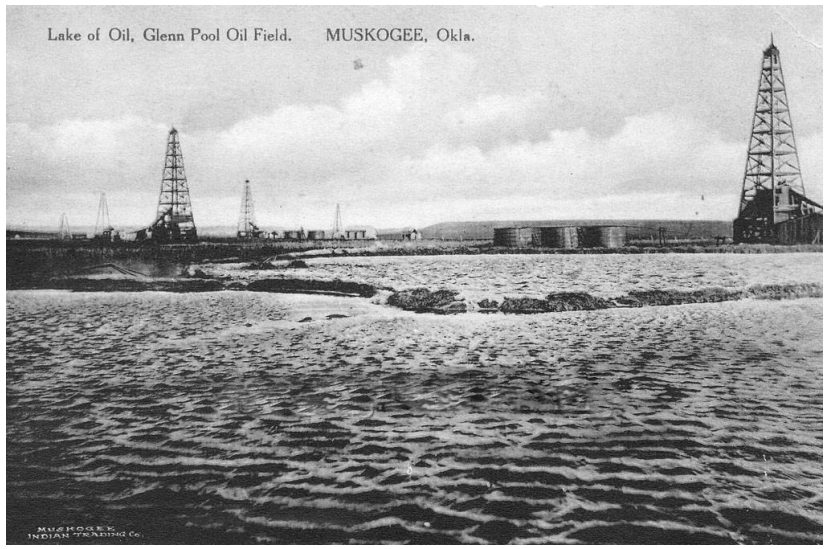



FIGURE 1.2: A pool of oil collected on the ground. Believed to be an emulsion. Postcard from Glenn Pool oil field, Oklahoma, c. 1907. (Postcard purchased and scanned by the author; in public domain due to age.)

Cottrell and Frederick "Buckner" Speed invented the electrocoalescer. The moment of invention is recounted by Speed[1] as inspired by Cottrell's work on removing smelter smoke from the air by electrostatic means. Speed brought up the topic of removing water from oil, and Cottrell replied:

> "Why, it's the same problem. For air put oil; for smoke particles, the minute water particles," and then his favourite form of expression: "What will happen if we put a high electrostatic stress on the oil?" Then in his characteristic quick manner, in a few minutes there were thrown together a beaker of oil, a spark coil, and two pieces of copper, and lo, the de-emulsification of the California oils had been solved. (Buckner Speed)

Their 1911 patent[2] is also an interesting read, particularly in how similar it is to modern day introductions to papers on the topic. In Figure 1.3 an extract from the drawings in the patent is shown, and again the similarity to modern equipment is interesting. We quote here from the patent introduction[2]:

> Much of the crude petroleum as it comes from the wells contains drops of water (...). In some cases this water settles out upon standing (...). But, at present, there still remains a large class of oils that cannot be economically freed from water without distillation. These are largely oils in which the water is in very small globules often less than one thousandth of an inch in diameter and behaving as if surrounded with a membrane resisting coalescence of the drops. (James Cottrell and Buckner Speed)

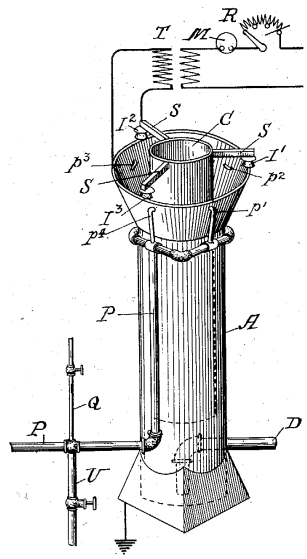



FIGURE 1.3: The first electrocoalescer design. A voltage is applied between two concentric pipes through which the oil flows. (From[2].)

In 1910, Allen Wright, working with Cottrell and Speed, formed the Petroleum Rectifying Corporation of California (PETRECO). By 1922 the company had 417 electrocoalescers in operation[3].

Meanwhile, on the other side of the country, a chemist named William S. Barnickel was working on the same problem. Visiting the Glenn Pool oil field (Figure 1.2) in 1907, and other fields in the area, he witnessed the destruction of the problematic oil[4]:

> At night (...) bright plumes of fire spewed from the ground, their intensity fueled by earthen pits of oil. During the day, the apocalyptic nighttime display gave way to billowing clouds of black smoke belching from the pits. (William S. Barnickel)

It goes without saying that the handling of roily oil was an environmental disaster. After four years of work, Barnickel discovered in late 1911 that an iron sulphate additive was very effective. After

[2] Cottrell, F. G. *et al.* (1911) [3] Bitto, R. *et al.* (2007) [4] Grant, T. (1996)

successful field trials in 1913, the patent was awarded in 1914. By 1922, the product named Tret-O-Lite was responsible for the yearly recovery of 50 million barrels of oil, which would otherwise have been disposed of[3]. In 1930, facing the threat of the great depression, the Tret-O-Lite company merged with PETRECO and formed the Petrolite Corporation, which remains today a major supplier to the oil industry (now as part of Baker Hughes). Figure 1.4 shows Mr. Barnickel standing with his foot on one of the first barrels of his demulsifier.

## §1.2 Scientific investigations of electrocoalescence

Published scientific investigations of the electrocoalescer and of removing water from crude oil begin in the second half of the 20th century. A landmark is the very descriptive 1965 paper on the electrocoalescer by Waterman[5], who had been working in Petrolite Corp. from 1926. This paper contains detailed schematics and explanations of the electrocoalescer. The year before, in 1964,

[3] Bitto, R. *et al.* (2007)


FIGURE 1.4: William S. Barnickel (left) posing with a barrel of the Tret-O-Lite product, in 1917. (Picture taken from Baker Hughes' Pinterest page, cropped and enhanced by the author, in public domain due to age.)

G. I. Taylor published a much more theoretical paper that helped lay the foundations for our detailed understanding of the electrocoalescer. Taylor[6] was concerned with a single drop, after experiments performed by himself and others about forty years earlier[7][8][9] on deformation and disintegration of drops in strong electric fields. Taylor derived an analytical expression for the deformation of a drop as a function of the dimensionless electric field strength, by considering the balance between electrostatic surface pressure and surface tension at the pole and at the equator of a drop. This formula is surprisingly accurate when compared to experimental results. The formula also predicts the critical electric field strength, above which drops disintegrate. Such disintegration produces many microdrops, which is obviously undesirable for electrocoalescers. Thus Taylor's result, while theoretical in nature, is very relevant to the application. Subsequent work by Brazier-Smith[10] improved on the Taylor result by considering the force balance not just at the pole and the equator, but on all the drop surface. The difference from Taylor's result is fairly small, and thus Taylor's result is in use still today. For a review of the then-nascent field of electrohydrodynamics, see Melcher *et al.*[11].

The results by Taylor on controlled drop disintegration laid the foundations for two important industrial processes, namely electrospraying and electrospinning. It is somehow fitting that electrospray ionisation has been used in mass spectrometry studies of the structural properties of asphaltenes.

Closer to the present time, a number of studies investigating systems important for crude oil processing have appeared in the literature. Among important contributions on removing water from crude oil we find the series of papers on "Dewatering of crude oil emulsions"[12][13][14][15] from a collaboration between Imperial College London and British Petroleum (BP) in the early 1990's. In the same period, the group of Sjöblom (at the same university as the author) became, and still is, very active in the area. In a *tour-de-force*, the group was to publish a series of 12 papers under the part-title "Water-in-crude oil emulsions from the Norwegian Continental Shelf"[16][17][18][19][20][21][22][23][24][25][26][27]. The perspective there is mainly from the chemistry point of view.

At my own department and at SINTEF Energy Research, studies of the electrocoalescer have been going on since 2002, with investigations of experimental and computational nature. In early

[7] Zeleny, J. (1917)
[8] Wilson, C. T. R. *et al.* (1925)
[9] Macky, W. A. (1931)
[10] Brazier-Smith, P. R. *et al.* (1971)
[11] Melcher, J. R. *et al.* (1969)
[12] Mohammed, R. *et al.* (1993)
[13] Mohammed, R. *et al.* (1993)
[14] Mohammed, R. *et al.* (1994)
[15] Chen, T. *et al.* (1994)
[16] Johansen, E. J. *et al.* (1989)
[17] Sjöblom, J. *et al.* (1990)
[18] Sjöblom, J. *et al.* (1990)
[19] Nordli, K. G. *et al.* (1991)
[20] Børve, K. G. N. *et al.* (1992)
[21] Mingyuan, L. *et al.* (1992)
[22] Sjöblom, J. *et al.* (1992)
[23] Urdahl, O. *et al.* (1993)
[24] Skodvin, T. *et al.* (1994)
[25] Sjöblom, J. *et al.* (1995)
[26] Rønningsen, H. P. *et al.* (1995)
[27] Urdahl, O. *et al.* (1995)

papers the turbulent flow of oil with water drops in an electric field was studied using Lagrangian particle tracking[28] [29], and the foundations were laid for the currently used two-phase flow simulations[30] [31]. In subsequent investigations, the coalescence of drop pairs and of a drop with a flat fluid interface was studied, both in experiments using real crude oils[32] and with simplified models[33] [34]. The simulation approach was concerned with the effect of electric fields and of surfactants on drop coalescence[35] [36] [37] [38]. Even more recently, studies have focused on single droplets and the interfacial properties of these, as considered in this thesis and contemporary experimental work.

Currently, many research groups are working on electrocoalescence in particular and crude oil emulsions in general, using different approaches with theory, simulation and experiments. But one point of interest is shared by essentially all groups, and that is the description of the complex interfaces of water drops in crude oil. It remains a major challenge to be able to size separator equipment based on pre-production samples from an oil well. Modern equipment is typically oversized by a significant factor. An improved understanding of how the water/oil interfacial properties affect the coalescer performance would lead to more compact and efficient electrocoalescers.

Even though the interfacial properties of water drops in oil have been under intense scrutiny, our understanding is still far from perfect. Recalling the patent application[2] by Cottrell and Speed quoted earlier, it has been known for a hundred years that the interface behaves in a qualitatively different way than typical fluid–fluid interfaces. Many theories have been put forward, implicating different chemical components of the crude oil. The asphaltenes have often been touted as the main cause of interfacial behaviour. Others have considered naphtenic acids, or that a particular subcomponent of the asphaltenes is responsible. For reviews on the role of crude oil components in emulsion stability, see e.g.[39] [40] [41] [42] [43]. Furthermore, there is the question of how these molecules arrange at the interface, possibly in an amorphous configuration together with other crude components such as resins.

[28] Chiesa, M. *et al.* (2005)
[29] Melheim, J. A. *et al.* (2006)
[30] Hansen, E. (2005)
[31] Bjørklund, E. (2009)
[32] Hellesø, S. *et al.* (2015)
[33] Atten, P. *et al.* (2006)
[34] Lundgaard, L. E. *et al.* (2006)
[35] Teigen, K. E. *et al.* (2009)
[36] Teigen, K. E. *et al.* (2010)
[37] Teigen, K. E. *et al.* (2011)
[38] Teigen, K. (2010)
[39] Jones, T. *et al.* (1978)
[40] Gafonova, O. V. *et al.* (2001)
[41] Sjöblom, J. (2001)
[42] Kokal, S. L. (2005)
[43] Kilpatrick, P. K. (2012)

## §1.3 Contributions of the present thesis

In this thesis, a multiscale simulation approach has been developed to test some hypotheses regarding the effect of various crude oil components on the interfacial properties. The approach combines coarse-grained molecular dynamics simulations, using the SAFT-$\gamma$ Mie approach, with two-phase flow simulations using a hybrid level-set/ghost-fluid/immersed-boundary method developed in this thesis. At the nanoscale, molecular dynamics simulations are used to obtain the interfacial tension $\gamma$ and the elasticity $K_a$ of oil-water interfaces covered with asphaltenes. These properties are used when computing the total tension in the macroscopic simulations. A case of particular interest is the crumpling of drops which are deflated by a needle. For this case, the simulation results show close agreement with experiments, both for very small drops manipulated with a micropipette and for larger drops attached to a pendant drop tensiometer.

The simpler system of water drops in a model oil with added non-ionic surfactant is also considered in this thesis. Using a combination of experiments and detailed two-phase flow simulations, we study water drops in oil being deformed by an electric field, and consider the effect of surfactant concentration in this system. It is found that the surfactant has an important effect on the damping of transient deformations, also when the surfactant concentration is so low that it does not significantly affect the interfacial tension.

Using theoretical considerations, we also investigate the system of a single spherical drop falling at steady state in the presence of surfactants. Eschewing the surfactant transport equation which is typically used in studies of this system, and employing instead the proper boundary conditions at the drop interface, we are able to obtain simultaneous analytical solutions to the velocity field and the interfacial tension profile along the drop. A simple model is introduced for the surfactant effect on the drop velocity, and it is shown that the predictions of this model match experimental results in the literature.

To enable future simulations of drop coalescence, we have developed a new curvature calculation method, called the LOLEX method, for the level-set interface-capturing method. The LOLEX method is robust to changes in the interfacial topology, for instance when two drops coalesce. The problem of computing the curvature e.g. at the moment when two drops are in close contact has been discussed previously in the literature. The LOLEX

method developed here is simpler and more general than previous approaches, and we demonstrate that it is easily extendable to three dimensions.

The two-phase flow simulations considered in this work are mainly performed in an axisymmetric coordinate system. To enable future simulations of fully three-dimensional flows, we have developed an MPI-parallel 3D version of the two-phase flow solver. Using the PETSc library for domain decomposition, the parallelised code shows good weak and strong scaling.

Finally, two computational tools have been developed, in order to increase the ease-of-use of molecular dynamics simulations with the SAFT-$\gamma$ Mie approach. Both of these tools are available as open source software.

## §1.4 Applications to other areas

Presently, it seems an inescapable fact that the production of crude oil will be significantly reduced in the coming years. In light of this, one might question the utility of scientific inquiry into processes related to this production. Fortunately, the methods and results developed in this thesis are transferable to studies in other fields. It is interesting to compare the present work with the recently defended PhD thesis of E. M. Kolahdouz[44], which also concerns the development of methods for simulating electrohydrodynamic deformations of elastic interfaces in two-phase incompressible flow. In contrast with the present work, his work is motivated by understanding the membranes of biological cells. Another example is the experimental technique of droplet-on-demand generation, where several approaches use electrohydrodynamic processes to create tiny droplets in a well-controlled fashion. Thus, both biological and chemical processes contain systems where phenomena arise similar to those seen in the electrocoalescer.

Further away from home, molecules very similar to the asphaltenes which are important in this work, namely polycyclic aromatic hydrocarbons, are important also in astrophysics[45]:

> Polycyclic aromatic hydrocarbons are abundant, ubiquitous, and a dominant force in the interstellar medium of galaxies.
> (Alexander G.G.M. Tielens)

[44] Kolahdouz, E. M. (2015)

[45] Tielens, A. G. (2008)

These molecules have also been identified as the major source of carbon in meteors[46], where they are found together with water ice. Who knows; perhaps the knowledge of crude oil/water separation gained over the past century will not become obsolete, but rather become an enabling technology in space exploration in centuries to come?

[46] Bernstein, M. P. *et al.* (1999)

CHAPTER 2

# Background

Contents


2.1 The molecular origin of fluid dynamics . . 11
2.2 Incompressible fluid flow . . . . . 18
2.3 Two-phase flow . . . . . 20
2.4 The effect of surfactants on two-phase flow 24
2.5 The asphaltenes . . . . . 27


> Because atomic behaviour is so unlike ordinary experience, it is very difficult to get used to and it appears peculiar and mysterious to everyone.
>
> Richard Feynman

IN THIS CHAPTER we will discuss the theoretical understanding of fluids, from the molecular forces, via the familiar partial differential equations describing fluid flow at the continuum scale, to the complex elastic interfaces. The history of the scientific and technological developments will also be reviewed where relevant. To provide a backdrop for the following sections, the incompressible Navier-Stokes equations, which tell us how the fluid flows, are given as

$$\nabla \cdot \mathbf{u} = 0, \tag{2.1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{\nabla p}{\rho} + \frac{\mu}{\rho}\nabla^2 \mathbf{u} + \mathbf{f}. \tag{2.2}$$

In these equations, the velocity vector field is denoted by $\mathbf{u}$, and the pressure field is denoted by $p$. The vector $\mathbf{f}$ is the acceleration due to any body force such as gravity, and it may be zero. The density and dynamic viscosity of the fluid are denoted by $\rho$ and $\mu$, respectively, and are assumed to be constant for each fluid.

## §2.1 The molecular origin of fluid dynamics

HOW THE MACROSCOPIC BEHAVIOUR of a fluid arises from the molecular forces has been a topic of scientific inquiry for

centuries. On the scale observable in the laboratory, fluids behave in agreement with equations that assume the fluid is a continuum. Yet we know that the fluid is not a continuum on the tiniest scale, where it is made up of molecules. Leucippus, Democritus and other Greek philosophers believed that matter must be made up by individual atoms (from Greek $\alpha\tau o\mu o\nu$, "indivisible"), but this remained a somewhat vague philosophical construction until Daniel Bernoulli's classic book *Hydrodynamica*[47] (which coined the term "hydrodynamics"). Here Bernoulli advanced the idea that gases are made up of atoms bouncing rapidly around. After Bernoulli, the atomic and molecular theories of matter in general, and fluids in particular, were developed to a much further degree. The iterations in the developments of the atomic and molecular theories are too numerous to recount here, but we will sketch the outlines of the derivation of the continuum equations from the molecular description of liquids, after having briefly discussed the differences between gas and liquid at the molecular scales, and the regimes of fluid flow where the continuum assumption is stretched to (and beyond) its limitations.

A good German translation exists of *Hydrodynamica*, due to Mikhailov et al.[48]. An English translation exists, by Carmody et al.[49], but it is not very good[50].

Depending on whether the fluid is gas or liquid, there are important differences in the molecular nature, while the continuum descriptions are more similar. As remarked by Maxwell in the Bakerian lecture delivered to the Royal Society in 1866 concerning his seminal work on the kinetic theory of gases[51] [52],

> The gaseous form of matter is distinguished by the great simplification which occurs in the expression of the properties of matter when it passes into that state from the solid or liquid form. (James Clerk Maxwell)

In contrast to this, a theoretical understanding of the nature of liquids from the atomic perspective has been much more difficult to attain. In addition to Maxwell's lecture, there are two other interesting Bakerian lectures that, taken together, shed light on this difference. In 1797, seventy years before Maxwell delivered his, Vince[53] gave the Bakerian lecture on "Experiments upon the resistance of bodies moving in fluids". In the introduction, he remarks that

> In the doctrine of the resistances of fluids, we see strong reasons to induce us to believe, that the theory cannot generally lead us to any true conclusions. When a body moves in a fluid, its particles strike the body; and, in our theoretical considerations, after this action, the particles are supposed to produce no further effect, but are conceived to be, as it were, annihilated. (Rev. Samuel Vince)

[47] Bernoulli, D. (1738) [51] Maxwell, J. C. (1866) [52] Maxwell, J. C. (1867)

In the Bakerian lecture delivered in 1962, almost a hundred years after Maxwell gave his on the gases, and over 160 years after Vince discards the theoretical approach as useless, Bernal[54] considers again our understanding of liquids from the basis of the atomic building blocks. In the introduction he remarks that

> It is admitted even by those who work most in the field that the study of the structure of liquids or any exposition of their properties in atomic terms is still largely to be sought. This is not for want of trying. (...) It is evident that any approach to a theory of liquids which aims at producing a picture of their structure and properties, will have to use very considerable approximations. (John D. Bernal)

In the half-century since then, progress from the theoretical side has remained slow. Fortunately, the "third way" of science, numerical simulation, has now become available to us, and has provided many new insights into the connections between the atomic and the continuum descriptions of fluids. Still to this day, simulations of atoms and molecules provide us with new insights into the behaviour of fluids; for instance in a very recently published paper, Nagata *et al.*[55] demonstrated using simulations that when water evaporates, there occurs a complicated dance involving three water molecules at the interface, which leads to one of the molecules going from the liquid to the gaseous state.

For the gases, as a consequence of their rarified nature, an important simplification comes from the existence of a well-defined criterion that tells us whether the assumption of continuum is valid for a particular system under study. This is the Knudsen number[56], which is defined as

$$Kn = \frac{\lambda}{L} \tag{2.3}$$

where $\lambda$ is the mean free path for a molecule in the gas, about 70 nanometres for air at standard temperature and pressure, and $L$ is some characteristic length scale, e.g. the pore size when gas flows through a permeable medium, or the distance between two colliding drops surrounded by gas. If the Knudsen number is much smaller than 1, the continuum hypothesis is valid. If we take as an example the case of two colliding drops surrounded by gas, the Knudsen number must become greater than 1 as the drops approach. If the time period in which $Kn \not\leq 1$ is very brief compared to the other important time scales, molecular effects are likely to be unimportant, whereas they will be important if this time period is longer. We find an example of this particular effect in the continuum simulations of drop collisions

[56] Knudsen, M. (1934)

by Pan *et al.*[57], who report very good agreement with experiments performed by Qian *et al.*[58], except for at very low Weber numbers (defined in Equation (2.14) in the next chapter), i.e. a longer period of $Kn \not\leq 1$, where the simulations predict behaviour qualitatively different from that seen in experiments.

For the liquid systems, there is no immediate analogue to the Knudsen number. Investigations of liquids at the molecular scale give behaviour similar to the continuum description for remarkably small-scale systems. E.g. in the study by Lau *et al.*[59], where the surface tension of water drops as a function of drop radius is calculated using molecular dynamics simulations, it is found that the surface tension is unaffected by the drop size all the way down to a drop radius of 1.5 nanometres (compared to the 0.28 nm diameter of a water molecule). Another striking example is the study of the capillary rise phenomenon at the molecular scale[60][61], where it is found that even for capillary tubes so small that the radius equals ten molecular diameters, the capillary rise behaviour follows relatively well the macroscopic Lucas-Washburn equation[62][63]. A review of the different regimes in gas and liquid at small scales is found in[64].

We will now proceed to give a brief overview of the path from the molecular to the continuum descriptions of fluid mechanics. We will necessarily be brief, for further details the reader is referred to the classic book by Chapman and Cowling[65].

At the scale of atoms and molecules, which we call the *nanoscale*, the governing equations are very different from Equations (2.1) and (2.2). At the nanoscale, atoms and molecules interact with each other through short-ranged forces. For simplicity, we consider a fluid that is made up of identical molecules, each consisting of a single atom. The force on one atom from another atom is given by the gradient of a potential that depends just on the separation $r$ between the two atoms. An example of this is the Lennard-Jones potential $V_{LJ}(r)$, illustrated in Figure 2.1 and given by the expression

This is not as theoretical as it sounds, but physically realisable with liquid argon, which has been a classic test for molecular dynamics simulations, and is well-described by the Lennard-Jones potential.

$$V(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]. \tag{2.4}$$

From this equation and from Figure 2.1, we see that the potential has an energy well, and the two parameters $\epsilon$ and $\sigma$ that enter into the expression for $V_{LJ}(r)$ determine the location and depth of this energy well. $r = \sigma$ is where the potential switches from being repulsive to

[60] Supple, S. *et al.* (2003)
[61] Dimitrov, D. I. *et al.* (2007)
[62] Lucas, R. (1918)
[63] Washburn, E. W. (1921)
[64] Gad-El-Hak, M. (2006)

being attractive, and $V_{LJ}(r)$ becomes large very quickly below $r = \sigma$, so $\sigma$ is often taken to be the diameter of the particle for visualisation purposes.

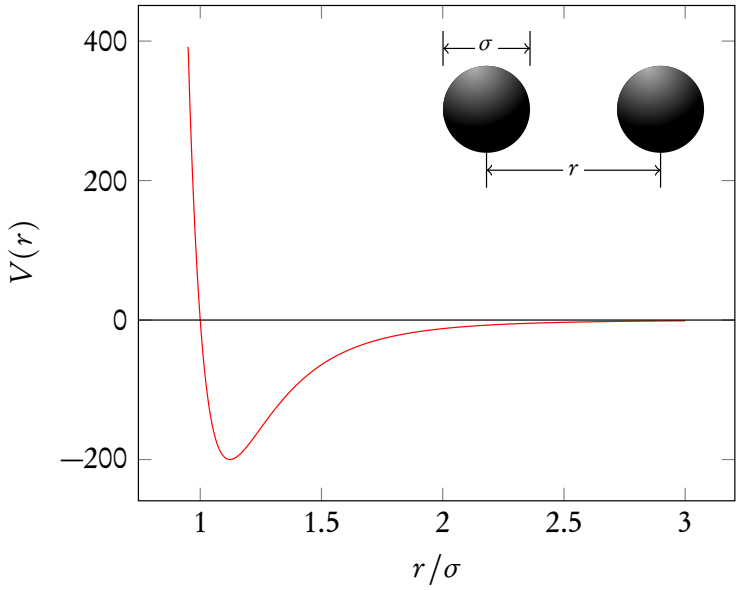



FIGURE 2.1: The Lennard-Jones potential plotted with $\epsilon = 200$ as a function of the dimensionless distance $r/\sigma$. Note that $r/\sigma = 1$ is where $V(r) = 0$, that $V(r) \to \infty$ as $r \to 0$, and that $V(r)$ quickly becomes very close to zero for increasing $r$. In the corner, the distance $r$ between particles and the "diameter" $\sigma$ is illustrated.

Given a system with $N$ such atoms in a three-dimensional space, all interacting by the forces just described, the state of a system at a given instant in time $t$ is described by $3N$ numbers that specify the positions $\mathbf{x}_i$ of all the atoms and $3N$ numbers that specify the momenta (i.e. mass $\times$ velocities) $\mathbf{p}_i$ of all the atoms. We can identify these $6N$ numbers as a single point in a $6N$-dimensional space, which is called *phase space*. To discuss how the system evolves in time, we can consider the $N$-particle distribution function $f_N(\mathbf{x}_1, .., \mathbf{x}_N, \mathbf{p}_1, .., \mathbf{p}_N, t)$ which is defined such that $f_N d^N x d^N p$ is the probability density of the system being in the infinitesimal phase-space volume element $d^N x d^N p$ at the instant $t$.

A celebrated result in classical mechanics is Liouville's theorem [66], which says that $f_N$ is constant along the trajectory that the system follows in phase-space. This is equivalent to the conservation of volume in phase-space, as expressed in the Liouville equation,

$$\frac{\partial f_N}{\partial t} + \sum_{i=1}^{N} (\partial_{\mathbf{x}_i} f_N) \cdot \frac{\partial \mathbf{x}_i}{\partial t} + \sum_{i=1}^{N} (\partial_{\mathbf{p}_i} f_N) \cdot \mathbf{F}_i = 0 \tag{2.5}$$

where $\partial_{\mathbf{x}_i}$ is the derivative with respect to the $i$'th atom's position, etc., and $\mathbf{F}_i$ is the force on atom $i$ from all the surrounding atoms. While it

[66] Liouville, J. (1838)

is far from immediately obvious that this description is useful, it will lead us on the path to Equations (2.1) and (2.2).

From the nanoscale, we must visit the intermediate *mesoscale* before we can ascend to the continuum realm. Transforming the nanoscale description of the fluid into the mesoscale is accomplished in two steps. In the first step we transform the single equation by Liouville into a chain of $N$ equations where equation $s$ in the chain connects the $s$-particle distribution function to the $s+1$ particle distribution function. This is called the Bogoliubov–Born–Green–Kirkwood–Yvon (BBGKY) hierarchy, after its authors[67] [68] [69] [70]. This transformation in itself does not make the problem any easier to solve. But by a remarkable approximation we can reduce the hierarchy of $N$ equations to just the single equation that relates the one-particle distribution function $f$ to the two-particle distribution function, replacing the two-particle distribution function with a closure relation which is just a function of $f$.

This approximation is what Boltzmann called the *Stoßzahlansatz*, introduced by Maxwell in the previously discussed work[52] on the kinematic theory of gases. The *Stoßzahlansatz* postulates that the initial velocities of two colliding atoms are uncorrelated. Boltzmann used this assumption to show that the entropy of a gas is nondecreasing. This approach was criticised by Loschmidt, who questioned how Boltzmann was able to arrive at an irreversible process by using equations and a formalism that was time-reversible. The resolution of this apparent paradox lies in the *Ansatz* itself: after collision, the velocities of the atoms must be correlated. By assuming that they are uncorrelated before the collision, time-asymmetry is introduced.

Notwithstanding the controversy, the *Stoßzahlansatz* can be used to simplify the BBGKY hierarchy into the Boltzmann equation,

$$\frac{\partial f}{\partial t} + (\partial_{\mathbf{x}} f) \cdot \frac{\partial \mathbf{x}}{\partial t} + (\partial_{\mathbf{p}} f) \cdot \mathbf{F} = \int \int (f' f_2' - f f_2) |\mathbf{n} \cdot \frac{(\mathbf{p} - \mathbf{p}_2)}{m}| d\mathbf{n} d\mathbf{p_2} \tag{2.6}$$

where the term on the right-hand side represents collisions; thus it is called the collision integral. In this term, $f'$ and $f_2'$ denote $f$ evaluated at the momenta $p, p_2$ of two atoms after a collision, and $f, f_2$ denote the same evaluated at the momenta of the atoms before a collision. $m$ denotes the atom mass. The collision angle vector $\mathbf{n}$ represents the

[67] Bogoliubov, N. N. (1946)
[68] Yvon, J. (1935)
[69] Kirkwood, J. G. (1946)
[70] Born, M. *et al.* (1946)
[52] Maxwell, J. C. (1867)

relative orientation of the two momenta, such that if we imagine two spherical atoms touching at the moment of collision, $\mathbf{n}$ points from the centre of one atom to the centre of another atom. (The magnitude of $\mathbf{n}$ is one and the sign is unimportant.)

From this we can proceed to the continuum description by the Chapman-Enskog procedure[65]. In this procedure the density $\rho$ and the continuum velocity $\mathbf{u}$ are obtained as the the velocity moments with respect to the mass and the momentum, respectively, of $f$. Thus, multiplying both sides of Equation (2.6) by the molecular mass and integrating over the velocity, one obtains

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

from which the Equation (2.1) is obtained by assuming incompressibility, i.e. $\boldsymbol{\nabla} \cdot \mathbf{u} = 0$. If we instead multiply both sides of Equation (2.6) by the momentum, and then integrate over the velocity, one obtains

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \mathbf{u}\mathbf{u}) - \boldsymbol{\nabla} \cdot \mathbb{T} = 0 \tag{2.8}$$

where $\mathbb{T}$ is the (Cauchy) stress tensor. The second term is simplified to $\rho(\mathbf{u} \cdot \boldsymbol{\nabla})\mathbf{u}$ by using the incompressibility approximation. To arrive at Equation (2.2) we must assume, in addition to incompressibility, that the fluid is Newtonian, and that the dynamic viscosity $\mu$ is constant throughout the fluid, which gives

$$\boldsymbol{\nabla} \cdot \mathbb{T} = -\boldsymbol{\nabla} p + \mu \boldsymbol{\nabla}^2 \mathbf{u}, \tag{2.9}$$

at which point we have arrived at the incompressible Navier-Stokes equations. For a much more thorough derivation of the last expression, see e.g. Chapter 3.3 in the book by Batchelor[71].

Higher-order approximations than the first moments of $f$, or equivalently approximations to higher than leading order in Knudsen number, lead to the Burnett or super-Burnett equations, as improvements over the Navier-Stokes equations. See e.g. the review by Agarwal *et al.*[72]. These equations are more complicated and have been investigated to a much more limited extent than the Navier-Stokes equations. Partly this is because the regime where the Burnett equations become important, $Kn \not\leq 1$, is important mostly for a limited number of applications which have been introduced by technological advances since the 1950's. Examples include the hypersonic flow around low-earth orbit spacecraft, analysis

[65] Chapman, S. *et al.* (1991)

of the deposition on spacecraft of exhaust from low thrust ion rocket engines, or rarified gas flow in micro-electro-mechanical systems (MEMS). Another complication with the Burnett equations is that stability issues arise when solving them numerically, so there has been much research into stabilising and regularising the Burnett equations[73][74][75][76]. As pointed out by Comeaux *et al.*[77], some of the origin of these difficulties lies in the fact that the Burnett equations are incompatible with the second law of thermodynamics.

We mention in closing that taking a separate path from the Boltzmann equation (2.6), namely simplifying the collision integral by using some collision model, leads to the numerical lattice-Boltzmann method which has become very popular in recent years. A common approach is the Bhatnagar-Gross-Krook (BGK) approximation[78] which replaces the right-hand side of Equation (2.6) by the much simpler term $\nu(f_0 - f)$, where $\nu$ is the molecular collision frequency and $f_0$ is the Maxwell-Boltzmann distribution, which is an analytical function of the particle mass, velocity and the temperature $T$.

## §2.2 Incompressible fluid flow

THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS, repeated below in Equations (2.10) and (2.11), have garnered a large interest by mathematicians and physicists alike for almost 200 years. They offer the tantalising combination of physical and industrial relevance (as they describe the motion of fluids) and a rich and complicated mathematical structure. It is an open problem, to the tune of $1 million from the Clay Mathematics Institute, to prove (or disprove) that solutions to these equations always exist and that the solutions have bounded kinetic energy. The equations are

$$\nabla \cdot \mathbf{u} = 0, \tag{2.10}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{\nabla p}{\rho} + \frac{\mu}{\rho}\nabla^2 \mathbf{u} + \mathbf{f}. \tag{2.11}$$

Another problem, for which the reward for a solution is implicit but significantly larger, is the efficient solution of these equations by a computer program. As an example, the computational fluid dynamics

[73] Zhong, X. *et al.* (1993)
[74] Jin, S. *et al.* (2001)
[75] Struchtrup, H. (2005)
[76] Söderholm, L. H. (2007)
[78] Bhatnagar, P. L. *et al.* (1954)

company CD-adapco, whose STAR-CCM+ software is among the more popular in the field, was recently acquired by Siemens for a price close to $1 billion. Computational methods used to solve these equations are discussed in Chapter 3.1.

We will in general restrict ourselves here to flows where the ratio of inertial to viscous forces is not too large. To be precise, we require that the Reynolds number $Re$ is smaller than some number,

$$Re = \frac{\rho |\mathbf{u}|_\infty L}{\mu} < Re_{\text{crit}}, \tag{2.12}$$

where $L$ and $|\mathbf{u}|_\infty$ are length- and velocity scales, respectively, which are characteristic of the system we are studying. We call this laminar flow. An example is the flow of honey poured onto your pancake at breakfast. $Re_{\text{crit}}$ is the critical Reynolds number, at which the transition from laminar to turbulent flow occurs. This number is different for different system. We remark that our assumption of incompressible flow, $\boldsymbol{\nabla} \cdot \mathbf{u} = 0$, is valid for flow speeds $|\mathbf{u}|_\infty < 0.3c$, where $c$ is the speed of sound. The number 0.3 is approximate, but the fact that this limit is roughly 100 m/s for air and much larger for most liquids indicates that incompressible flow is relevant for a large number of situations.

In much of the work we are concerned with here, the Reynolds number is so low that the flow is not only laminar, but indeed what is called *creeping*, which occurs for $Re \ll 1$. This is generally the case for drops in an emulsion, which are of a very small size and immersed in a very viscous outer fluid, two factors which conspire to reduce the Reynolds number. In this regime, some of our physical intuition with fluids breaks down. Perhaps the best system to consider to increase our intuition is when breaking eggs into a bowl (perhaps when making the aforementioned pancakes): if a tiny piece of egg shell falls into the bowl and settles at the bottom, picking it up can be very difficult. It almost seems as if the egg shell moves away from one's finger, but in reality it is the fluid boundary layer around the finger which pushes the shell away. The chance of success is greatly increased by moving the finger at an excruciatingly slow speed; one has then lived, but for a moment, in a situation governed by very low Reynolds number flow.

Other organisms, much smaller than ourselves, spend their entire lives at $Re \ll 1$. "Life at low Reynolds number" is the title of the very interesting and pedagogical 1977 paper by Purcell[79], from which we shall quote in the following. While turbulent flows are of great

importance and visually very impressive, low Reynolds number flows are now coming into their own. In the past decade there has been an explosion of interest in swimming at low Reynolds numbers, which is relevant for many biological systems and has coined the term biological fluid mechanics. Purcell, recipient of the 1952 Nobel Prize in physics for the discovery of nuclear magnetic resonance (NMR), was in some of his later work inspired by his friendship with the physicist-turned-biologist Howard Berg. Purcell had the prescience to anticipate the recent development of biological fluid mechanics in the late 1970's.

Looking at the number of citations of Purcell's paper gives an impression of the recent rise in biological fluid mechanics: up until the year 2001, the paper was cited about 10 times every year. Then from 2001 to 2009 the citation frequency increases by a factor of 10, and has now remained above 100 citations per year since 2009.

To further illustrate the situation in very low Reynolds number flows, we quote here from Purcell's paper[79], where he discusses a $\sim 2\mu$m particle (or bacteria) moving in water at a typical speed of $30\mu$m/s. This is quite similar to the situation for sedimenting emulsion drops.

> If I have to push that animal to move it, and suddenly I stop pushing, how far will it coast before it slows down? The answer is, about 0.01 nanometres. And it takes about 0.6 microseconds to slow down. I think this makes it clear what low Reynolds number means. Inertia plays no role whatsoever. If you are at very low Reynolds number, what you are doing at the moment is entirely determined by the forces that are exerted on you at that moment, and by nothing in the past. (Edward M. Purcell)

In more mathematical language, very low $Re$ simplifies Equation (2.2) to the Stokes equation,

$$\frac{\mu}{\rho}\nabla^2\mathbf{u} - \frac{\nabla p}{\rho} + \mathbf{f} = 0 \tag{2.13}$$

which as Purcell notes has the property of instantaneity: the flow at a single point in time does not depend upon the flow in the past. However, we can produce a time-varying flow by having the body force (or the boundary conditions) be a function of time. Even in the presence of such time variations, the flow is instantaneous, and therefore it is reversible. A striking visualisation of this is the reversible "mixing" of dye drops in a cylindrical Couette flow, see e.g. the video at http://youtu.be/_dbnH-BBSNo.

## §2.3 Two-phase flow

THE EQUATIONS (2.1) AND (2.2) above are valid for single-phase flow. To extend this formulation to two-phase flow, we keep these

equations in each of the two phases, where the densities, viscosities etc. are constant in each phase. We will denote the properties of the fluid in a drop with the subscript $_1$, and the properties of the surrounding fluid with the subscript $_2$. Across the interface between the fluids, a jump in e.g. pressure will arise mainly due to interfacial tension, and this together with effects of the jump in density and viscosity must be added to our equations. When interfacial tension is involved, a second important nondimensional quantity is the Weber number,

$$We = \frac{\rho |\mathbf{u}|_\infty^2 L}{\gamma} \tag{2.14}$$

where $\gamma$ is the interfacial tension. The Weber number describes the ratio of inertial to interfacial forces; at small We, the shape of a droplet will remain close to spherical, while at large We its shape may become arbitrarily distorted.

Since the work presented in this thesis is predominantly occupied with droplets, droplets will also be the focus in this section. Good references on droplet behaviour include the book by Clift *et al.* [80], as well as several chapters in the book by Levich [81] (see also Leal [82]). There are many other important categories of two-phase flow, for instance waves at sea or the flow of two fluids in a pipe. For general texts on two-phase flows, the reader may consult Levy [83] for gas-liquid flows and Brauner [84] for liquid-liquid flows.

Coming back to our focus on droplets, there are two very important phenomena to consider. One is the oscillation of a droplet immersed in another fluid, and the other is the fall of a droplet through another fluid. We begin by considering the oscillating fluid drop.

The first published investigation of oscillating drops is by Bidone [85] who in 1829 published the results of experiments with the breakup of thin liquid jets, and who also considered the oscillating drops that are formed after this breakup. Some of those results are reproduced by Rayleigh [86] in his seminal study, where a theoretical analysis is presented that explains the breakup phenomenon. This instability is today known as the Rayleigh instability. The paper also discusses the oscillations of the remaining drops, and a formula is derived for the oscillation frequency, valid for small deformations and neglecting the external fluid and the viscosity of the drop.

Later work by Lamb [87], Chandrasekhar [88], Miller *et al.* [89] and Prosperetti [90] extended the result, first to a weakly viscous drop, then to any drop viscosity, then to including also an external fluid, and

finally to considering also the initial-value problem (as opposed to the asymptotic oscillations). These theoretical results have been applied to the measurement of interfacial tension and viscosity[91], but it is important to note that, as shown by Prosperetti[90], the classical and easily tractable results are only valid after at least one oscillation cycle has passed. This is because the initial value problem of e.g. a stretched drop at rest is not in the solution space of the classic results. The results by Prosperetti give solutions to the initial value problem, but for them to be useful a numerical inverse Laplace transform must be applied. Thus these results are not much used in the literature.

Turning now to the falling drop, the story becomes even more interesting. The simplest case to consider is the perfectly spherical drop falling at terminal velocity, and where the drop is small enough (relative to the external fluid viscosity) that $Re \ll 1$. As discussed previously, this simplifies the Navier-Stokes equations significantly. Thus for this important case, Hadamard[92] and Rybzynski[93] were able in 1911 to obtain (independently) the analytical solution to the entire flow field, both inside and around the drop. But alas, the terminal velocity predicted by their result,

$$v_{\mathrm{T,HR}} = \frac{(\rho_1 - \rho_2)\mathbf{g}D^2(\mu_1 + \mu_2)}{6\mu_2(3\mu_1 + 2\mu_2)}, \tag{2.15}$$

does not agree with experimental results for small falling drops. The experiments rather appear to agree with the formula given earlier by Stokes[94] for the terminal velocity of a falling solid sphere at low $Re$,

$$v_{\mathrm{T,S}} = \frac{(\rho_1 - \rho_2)\mathbf{g}D^2}{18\mu_2}. \tag{2.16}$$

This disagreement is even pointed out by Hadamard himself in the concluding remarks in his 1911 paper:

> La formule (III) présente, avec les résultats expérimentaux obtenus quant à présent (et encore inédits), de notables divergences. Il semble donc, jusqu'á nouvel ordre, que, dans les cas étudiés, les hypothèses classiques dont nous sommes parti doivent être modifiées. (Jacques S. Hadamard)

Translated: "The formula (III) gives notable disagreement with experimental results obtained recently (and not yet published). Presently it appears that, for the cases considered here, the assumed classical hypotheses require modifications."

Explaining this discrepancy attracted considerable attention during the 20th century. Many papers[95] [96] [97] [98] [99] presented careful experiments showing the discrepancy for a number of different fluid combinations; we mention among these air in water, mercury

[91] Ronay, M. (1978)
[95] Nordlund, I. (1913)
[96] Lebedev, A. (1916)
[97] Silvey, O. W. (1916)
[98] Bond, W. N. (1927)
[99] Bond, W. N. *et al.* (1928)

in castor oil, water in castor oil, etc. Only by increasing the droplet radius was one able to obtain the terminal velocity predicted by Equation (2.15). An early hypothesis by Boussinesq[100] was that a thin but macroscopic layer close to the interface has a significantly enhanced viscosity, which retards the droplet motion. This explanation was accepted for many years, but was eventually discarded in favour of the currently prevailing notion[80][81], namely that trace contaminants present in one (or both) of the fluids act as surfactants and modify the interfacial tension. When the droplet falls, the flow transports the surfactants towards the upper stagnation point. This causes a Marangoni force[101][102] which attempts to restore the uniform surfactant concentration. As a result, the velocity along the interface vanishes (in the absence of surfactant diffusion). The interfacial diffusion coefficient of surfactants is hard to estimate, but it is in general a very low number, $O(10^{-7})$ m/s$^2$[103], meaning one can get very close to the terminal velocity of a hard sphere. To explain that this effect vanishes for increasing droplet radii, the stagnant cap model was introduced. This model posits that the rear cap of the interface is immobile due to the surfactant, while the front cap (i.e. the complement of the rear cap) is fully fluid. By specifying a criterion that varies the cap angle as a function of the drop radius, one can obtain good agreement with experimental results.

In parallel to the Western investigations, Soviet researchers made great (perhaps even greater) progress on understanding the problem. Volarovich *et al.*[104] were the first to perform experiments where good agreement with the Hadamard-Rybczynski result was observed. The fluid combination was exotic to say the least, namely molten lead in molten boron trioxide. One is perhaps not surprised that surfactants were not present in this system; it is indeed difficult to envision a surfactant molecule that would have an effect on these fluids. Later experiments with slightly less exotic fluids include those by Frumkin *et al.*[105] using mercury drops in highly purified glycerine. A detailed study was made by Frumkin *et al.*[106] on the effect of surfactants on falling drops. The discussion in the book by Levich[81] (who was a student of Frumkin) is quite illuminating. Fortunately, Western researchers (among them Acrivos) had realised that Levich's research was interesting, and put down a significant effort to translate the book into English. One can but lament that this translated edition has not been republished since 1962; and that yet other works remain available only in Russian, e.g. the book "Coagulation and Dynamics of Thin

[80] Clift, R. *et al.* (2013)
[81] Levich, V. G. (1962)
[101] Marangoni, C. (1865)
[102] Marangoni, C. (1870)
[103] Sakata, E. *et al.* (1969)

Films" by Dukhin *et al.*[107]. More easily available is the review paper by Levich *et al.*[108] in the inaugural issue of Ann. Rev. Fluid Mech., which also provides a readable introduction to a wide array of surface tension-related phenomena.

The case of the falling drop will be revisited in Chapter 5.5, where a new model is presented as part of this work to explain the transition in terminal velocity as the drop size increases. This new model is an alternative to the currently accepted stagnant cap model, and we highlight the main differences between the two. We point out that the stagnant cap model as applied to surfactants is not internally consistent, and discuss the situations where the two models are applicable.

## §2.4 The effect of surfactants on two-phase flow

HAVING CONSIDERED the oscillating and the falling drop cases, we finally give some remarks on surfactants from the chemical perspective. Surfactants have been the subject of much research, since they are ubiquitous in nature, for instance in the processes in our lungs, and important in chemical processes particularly in the food and cosmetics industries. What is likely the first mention of the intentional use of surface-active compounds to alter the interfacial properties of liquids we find in the writings of Pliny the Elder. In book two of his *Naturalis Historia*[109] [110], published in 77 AD, he mentions among other facts about liquids the use of oil to calm the sea:

> Omne oleo tranquillari, et ob id urinantes ore spargere, quoniam mitiget naturam asperam lucemque deportet.
>
> All sea water is made smooth by oil, and so divers sprinkle oil from their mouth because it calms the rough element and carries light down with them.
>
> (Pliny the Elder)

That oil calms the surface of the sea was rediscovered by Benjamin Franklin, who had read Pliny's account but failed to heed it. During one of his voyages by ship, Franklin was scoffed at by the captain when he expressed curiosity as to why the sea was calmer behind some ships; evidently these had been discharging grease from the galley out of the scuppers, which the captain thought any man worth his salt should be able to surmise. Franklin[111] subsequently undertook the first proper experiments with surface-active agents,

[109] Pliny the Elder (1906) [110] Pliny the Elder (1949)

conducting his experiments on the pond in Clapham Common, a park in south London. An interesting popular account is given in the book by Tanford[112]. Later work by Rayleigh[113][114], Pockels[115], Plateau[116] and Marangoni[117], to mention but a few, expanded our understanding of surfactants acting on liquid interfaces.

Of particular note here is the German chemist Agnes Pockels (1862–1935), who never received any formal scientific training (being a woman), but nevertheless became a pioneer in the study of surfactants and fluid interfaces. When cleaning dishes (!) Pockels observed some curious effects of soaps and greases, and over the course of a few years invented a first version of the instrument we today know as the Langmuir-Blodgett trough. With this she measured not only the surface tension of the films she observed, but she also studied the effect of stretching and compressing the films, the effects of the type of contaminant, the effect of additives to the bulk phase, and made deductions about the amount of material adsorbed at the interface. In later works she also estimated the thickness of these films with remarkable precision. Through her brother, the physicist Friedrich Pockels, she was able to obtain Lord Rayleigh's writings on oil films on water, and subsequently wrote letters to him on the phenomena she herself had observed. Rayleigh was so impressed with these results that he assisted Pockels in publishing them in Nature[115]. She subsequently published several important results in surface chemistry: she was the first to plot quantitative surface pressure isotherms[118], and she made significant improvements to the ring tensiometer[119]. In 1931 she received the Laura-Leonard prize of the German *Kolloid-Gesellschaft*, and when she turned 70 the year later she received an honorary doctorate from the University of Braunschweig. Building on (and acknowledging) Pockels' work, Langmuir and Blodgett made improvements to the trough apparatus, and Langmuir's research on the properties of surface films earned him the 1932 Nobel Prize in chemistry. In the same year, Pockels was quoted by Ostwald in his *Festschrift* in her honor[120], saying

> I learned to my great joy that my work is being used by others for their investigations. (Agnes Pockels)

The interested reader is referred to the book by Roberts[121] for further historical details and descriptions of the trough techniques.

In the 20th century, a large body of research on surfactants has been produced, not in the least by the group at the Unilever research laboratory in the Netherlands, starting in 1960 with van den Tempel[122]. Subsequent work up to the present day by van den

[114] Rayleigh, L. (1890)
[115] Pockels, A. (1891)
[118] Pockels, A. (1893)
[119] Pockels, A. (1926)
[120] Ostwald, W. (1932)

Tempel, Lucassen-Reynders *et al.*[123], Lucassen[124] and others has significantly improved our understanding of the effects of surfactants on fluid interfaces in general and on emulsions in particular. In this context, elasticity as defined by Gibbs[125] is very important for the emulsion stability. Methods such as the previously discussed troughs, or the Wilhelmy plate, have traditionally been used to investigate elasticity. But in recent years the pendant drop tensiometer has become the preferred instrument. An overview of such measurement techniques is found in the recent review by Fuller *et al.*[126].

We remark that the static conditions in such tensiometers are very different from the flow in an emulsion being separated. As an example, while the diffusion boundary layer for a surfactant can easily become comparable to the drop radius in the absence of flow, leading to time scales of hours or even days to obtain equilibrium, the same equilibrium can be reached in a matter of seconds for a falling drop due to the accelerating factor given by the Schmidt number $Sc = \nu/D$. Here, $\nu = \mu/\rho$ is the kinematic viscosity of the fluid and $D$ is the diffusion coefficient of the surfactant, typically $O(10^{-10})$, so the Schmidt number can quickly be of the order of $10^6$. With this in mind, some care should be taken when comparing the equilibrium situations in these two systems, since in the former case there is ample time for reorientation processes at the interface to take place, which may not be true in the latter case.

As another point to note, it is questionable whether the same mechanisms (i.e. a Gibbs-type of elasticity) are responsible for emulsion stability in water-crude oil emulsions. For instance it has been noted for the Gibbs elasticity found in surfactant-stabilised systems that[127]

> The characteristic maximum in the elasticity is not caused by any molecular interactions in the surface, but purely by the effect of surfactant supply to the surface from the solution at higher concentration. Molecular interactions could not play a part in this, in any case, because Langmuir adsorption presupposes ideal surface behaviour.
>
> (Emmi H. Lucassen-Reynders)

Thus the emulsion stability mechanism discussed in the context of surfactant-stabilised emulsions is qualitatively different from the stabilising mechanism of crude oil components like asphaltenes, which are believed to adsorb essentially irreversibly at the interface, and which have a decidedly non-ideal surface behaviour.

[127] Lucassen-Reynders, E. (1996)

## §2.5 The asphaltenes

WITH THESE DIFFERENCES IN MIND, let us then consider the hitherto-elusive asphaltenes, whose fundamental properties are debated still today. Crude oil is a complex mixture of hydrocarbons, ranging from the simplest that exist (methane, $CH_4$) to the complex and heavy asphaltenes, having a molecular weight roughly 50 times that of methane. These components are separated from each other by depressurisation, distillation and other refinement processes. The only workable definition that exists of asphaltenes is as a solubility class; for instance as codified in the ASTM D 6560 – 00 standard where it is defined as the component of crude oil that is soluble in toluene but insoluble in heptane. So we know how to isolate asphaltenes, but precisely *what they are* is still up for debate. Mullins quips in the preface to the detailed overview given in *Asphaltenes, Heavy Oil and Petroleomics*:

> To all those scientists and technologists who have and will become enthralled and enchanted by the wiles of the asphaltenes and heavy oils, and to the families and friends of our fold who at least feign enthusiasm when subjected to renderings of the mysterious objects of our study.
>
> (Oliver C. Mullins)

In addition to being enigmatic, asphaltenes are also highly problematic. Their mischief begins in the reservoir, where they may be deposited on the walls of the porous rock around the wellhead, decreasing the permeability and with it the flow of oil. Further downstream, they can also be deposited on pipeline walls and cause complete blockage of pipelines tens of centimetres in diameter; this has earned them a reputation as the "cholesterol of crude oil". As the flow proceeds to the topsides, it comes to the separator, where asphaltenes cause problems by significantly increasing emulsion stability and increasing the water content of the final produced oil, necessitating larger and more complicated separation facilities. When finally the crude oil has been processed such that it is of export quality, it is pumped into ships or pipelines and transported to onshore refineries where the oil is converted into useful products. Here again the asphaltenes rear their heads, fouling up heat exchangers and causing problems if oils with too different properties are mixed. Yet in the end, the asphaltenes are also useful to us, since they are (as the name suggests) used in the production of the asphalt used to pave our roads.

It is in the context of adhesion materials that we first encounter the asphaltenes. The term was coined by Boussingault[128] in 1836. In a

striking similarity to the modern operational definition, Boussingault noted that the asphaltenes, which he had produced as the remainder after distilling bitumen, were insoluble in alcohol and soluble in turpentine (an aromatic solvent). Hoepfner[129] has a delightful historical review of the asphaltenes in Chapter 1.1 of his thesis. It is interesting to note that the asphaltenes have been the subject of (at times heated) debate in the literature since 1893. Still today there are many questions about the asphaltenes that have not been fully resolved: How do the molecules themselves typically look? How do they aggregate at different concentrations, temperatures and pressures? Are different subfractions of the asphaltenes more or less responsible for different phenomena? How do they interact with other crude fractions such as resins? Insofar as there exists a consensus in the literature today, it is represented by the Yen-Mullins model[130] [131] [132] [133]. While this thesis was going through final preparations for printing, a paper by Evdokimov *et al.* [134] appeared in the literature, claiming that the Yen-Mullins model is "fundamentally wrong". It appears likely that the debate on asphaltene structure and behaviour will continue for years to come.

In the Yen-Mullins model, the individual asphaltene molecules are of the "continental" type, i.e. a central core is made up of aromatic rings, and on the periphery there are attached aliphatic chains, with heteroatom substitution being common. Beyond this overall picture there is great uncertainty. The most advanced experimental techniques available to modern physicists and chemists have been applied to the study of the asphaltenes, such as neutron scattering from the highest intensity sources, or high-resolution mass spectrometry using increasingly sophisticated techniques. Very recently, state-of-the-art atomic force microscopy (AFM) and scanning tunneling microscopy (STM) methods from the IBM Research group in Zürich have given us the first directmolecular images of asphaltene molecules[136]. Images of one such molecule are reproduced in Figure 2.2 on the next page. Taken together, these investigations have provided us with a picture of asphaltenes as one of the most complex materials ever studied[129].

It is interesting to note that analysis of crude oils is claimed[135] to have directly caused mass spectrometry techniques to become the popular tool they are today in all areas of analytical chemistry.

In a given sample of asphaltenes, experiments indicate that there are thousands of different empirical chemical formulas. Each of these represents a huge class of molecules; take for instance the formula $C_6H_{12}O_6$, which is not an asphaltene, but a more familiar simple sugar molecule that forms a single pentagonal or hexagonal ring. This

[130] Dickie, J. P. *et al.* (1967)
[131] Mullins, O. C. (2010)
[132] Mullins, O. C. (2011)
[133] Mullins, O. C. *et al.* (2012)
[136] Schuler, B. *et al.* (2015)
[129] Hoepfner, M. P. (2013)

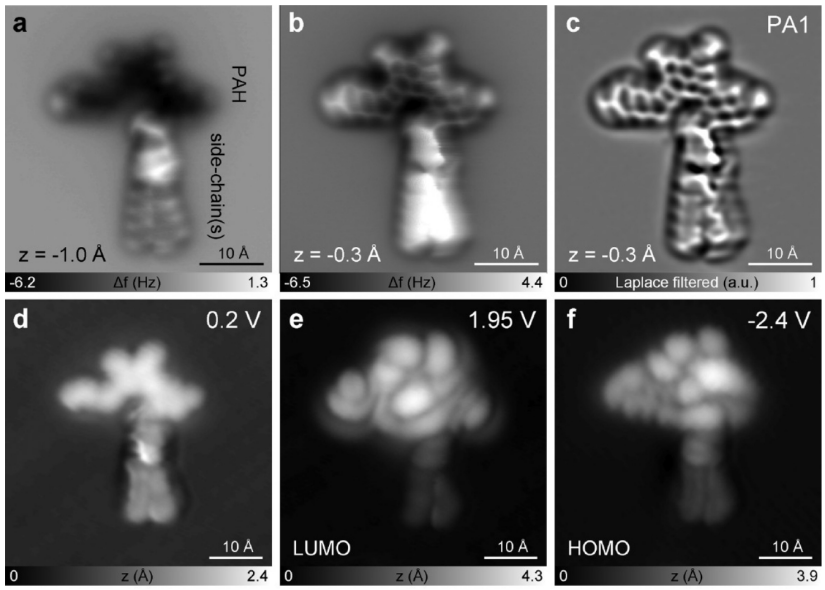



FIGURE 2.2: An asphaltene molecule imaged with AFM (top row) showing the molecular structure, and with STM (bottom row) showing molecular orbital isosurfaces. The top half of the molecule is an aromatic core and the bottom half appears to be two aliphatic tails. (Reprinted from Schuler, B., Meyer, G., Peña, D., Mullins, O. C. & Gross, L. Unraveling the Molecular Structures of Asphaltenes by Atomic Force Microscopy. *Journal of the American Chemical Society*, **137**, 9870–9876. doi:10.1021/jacs.5b04056 (2015). Copyright (2015) American Chemical Society.)

formula may refer to any one of tens of different monosaccharids like fructose, glucose, galactose, mannose, sorbose etc. Since the asphaltenes have not six but of the order of 50 carbon atoms, forming not one but typically six or seven rings with aliphatic tails attached, and with heteroatoms like nitrogen, sulfur or oxygen occasionally substituted in the ring structures[137][132][133][138][139][136][140], we understand that the number of isomers for a given empirical formula is in the trillions. A good analogy to asphaltene molecules is snowflakes; according to the old adage, no two are exactly the same.

As alluded to above, the effect of asphaltenes on liquid-liquid interfaces (and thus on separation processes) is also complicated. There are many reviews in the literature on the stability of water-crude oil emulsions[39][40][41][42][43]. The interfacial properties of drops covered with asphaltenes produce curious phenomena, such as the crumpled drops seen in experiments by Yeung *et al.*[141].

In the present work, it is attempted to obtain the mechanical properties of asphaltene-covered liquid-liquid interfaces from molecular dynamics simulations, and the implications of these properties are then studied using continuum simulations of two-phase flow with

[137] Goual, L. *et al.* (2002)
[132] Mullins, O. C. (2011)
[133] Mullins, O. C. *et al.* (2012)
[138] McKenna, A. M. *et al.* (2013)
[139] Podgorski, D. C. *et al.* (2013)
[136] Schuler, B. *et al.* (2015)
[140] Tang, W. *et al.* (2015)
[39] Jones, T. *et al.* (1978)
[40] Gafonova, O. V. *et al.* (2001)
[41] Sjöblom, J. (2001)
[42] Kokal, S. L. (2005)
[43] Kilpatrick, P. K. (2012)

corresponding properties. A significant advantage of the molecular dynamics approach over an experimental approach is the possibility of detailed hypothesis testing with regards to e.g. the asphaltene structure and the effect of other crude components. Another advantage is that it gives direct insight into how the molecules behave at the interface. The main disadvantage is, as with any simulation tool, that one does not know *a priori* that the simulation is representative of reality. With this in mind, we proceed to consider the computational methods used in the present work.

CHAPTER 3

# Computational methods

Contents


3.1 Methods for incompressible flow . . . . . 31
3.2 Methods for two-phase flow . . . . . 42
3.3 Methods for elastic interfaces . . . . . 47
3.4 Methods for molecular simulation . . . . . 52


> Wind tunnels won't go away – we'll need them to store the printouts from our CFD solutions!
>
> Dean R. Chapman (joking)

## §3.1 Methods for incompressible flow

When solving partial differential equations (PDEs) using the discrete memory of a computer, our solution will be given as a set of values on a finite number of points in space-time. The way we move our continuous problem over to a discrete space-time is called a discretisation. For the spatial discretisation we consider here finite difference methods; a simple example is the central difference formula:

$$\frac{\partial u(x)}{\partial x} \approx \frac{u(x+\frac{1}{2}\Delta x) - u(x-\frac{1}{2}\Delta x)}{\Delta x}. \tag{3.1}$$

Several advanced schemes are readily available, see e.g. Jiang *et al.* [142] [143] for the popular WENO schemes.

Given the Navier-Stokes equations (2.1) and (2.2), several schemes have been devised to solve these on a computer in a fast manner, for various definitions of "fast". Our basis here will be finite difference methods on a structured rectangular uniform grid. It is common to employ a staggered grid to avoid checkerboarding of the pressure field; this means that the pressure and other scalars "live" at cell centres, while the velocities "live" at the cell faces. To be more precise, if we have a pressure at one point $p_{i,j,k}$, the velocities around this point

are (taking $\mathbf{u} = [u\,v\,w]^T$) $u_{i\pm1/2,j,k}, v_{i,j\pm1/2,k}, w_{i,j,k\pm1/2}$ located at the 6 cell faces. See Figure 3.1 for a schematic of a staggered grid in two dimensions. A technical detail regarding the staggered grid, which can be seen in Figure 3.1, is that one has a choice of the number of points used to discretise e.g. the $u$-component in the $x$-direction. If there are $N_x$ points used to discretise the pressure (and other scalars) in the $x$-direction, there must be either $N_x+1$ or $N_x-1$ points used to discretise $u$ in the $x$-direction (and similarly for other vectors). In the present work we choose to use $N_x-1$. This is illustrated in Figure 3.1, where 5 pressure values are given in the $x$-direction, while 4 velocity values (together with two ghost values at the boundaries, shown in red) are given. Note that no ghost points are needed for the pressure, since the pressure boundary condition is zero Neumann, as will be discussed in the following. Note also that instead of a staggered grid, one could use a colocated grid together with some form of interpolation, as popularised by Rhie and Chow [144].

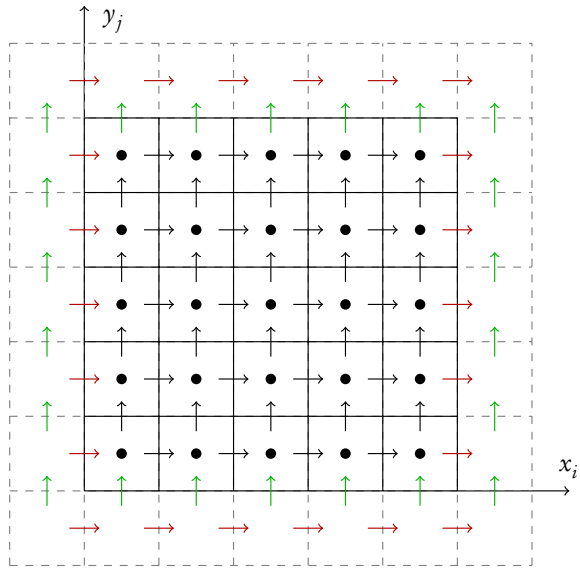



FIGURE 3.1: Illustration of the staggered grid in two dimensions. The pressure is given at the cell centres (black points). The $u$ and $v$ velocity components are offset in the $x$- and $y$-directions, respectively. Boundary conditions are shown in red for the $u$ component and green for the $v$ component. No boundary conditions are given for the pressure.

Incompressibility is both a blessing and a curse. Physically, no material can be absolutely incompressible, so it is an approximation. In many cases, and particularly for liquids, it is a very good approximation. In an incompressible fluid, the speed of sound is infinite, so there are no sound waves. This is beneficial, since sound waves are typically not relevant for the phenomena we are interested in. For compressible flows, much research has gone into making good numerical methods for low Mach number flows, i.e. in the case where the sound waves are not of interest but may obscure the phenomena

under study, and also make the simulations slow since one has to resolve waves that are much faster than the phenomena under study. In this sense, incompressibility makes life easier.

The back side of the coin is that, since sound speed is infinite for an incompressible fluid, all points in the simulation domain are affected by all other points in each time step. This puts limitations on how well the method scales to many CPU cores, since communication must occur between all grid cells, and communication between cores is a bottleneck.

As a concrete example, we consider here the widely used projection method due to Chorin[145]. This is a form of splitting method. Mathematically, it relies on the Helmholtz-Hodge decomposition theorem; while interesting in its own right, we will not discuss the theorem and its application to the projection method here. The interested reader is referred to the survey by Bhatia *et al.*[146].

Starting with Equation (2.2), we replace the time derivative $\frac{\partial u}{\partial t}$ by $(\mathbf{u}^{n+1} - \mathbf{u}^n)/h$, as in Euler's method with a discrete time step $h$. Next we introduce an intermediate velocity field, $\mathbf{u}^*$, which will be a solution in between the two time-steps,

$$\rho \frac{\mathbf{u}^{n+1} - \mathbf{u}^* + \mathbf{u}^* - \mathbf{u}^n}{h} = \mathbf{a}(\mathbf{u}^n) - \nabla p^{n+1}. \tag{3.2}$$

Here $\mathbf{a}(\mathbf{u}^n)$ contains all terms apart from the time derivative and the pressure gradient in Equation (2.2). We are free to choose $\mathbf{u}^*$ in Equation (3.2), and we may in particular choose it such that

$$\rho \frac{\mathbf{u}^* - \mathbf{u}^n}{h} = \mathbf{a}(\mathbf{u}^n), \tag{3.3}$$

$$\rho \frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{h} = -\nabla p^{n+1}. \tag{3.4}$$

With this choice we calculate $\frac{\rho}{h}\mathbf{u}^*$ from Equation (3.3), and it only depends on the values at time step $n$,

$$\frac{\rho}{h}\mathbf{u}^* = \mathbf{a}(\mathbf{u}^n) + \frac{\rho}{h}\mathbf{u}^n \tag{3.5}$$

If we take the divergence of Equation (3.4) and use that $\nabla \cdot \mathbf{u}^{n+1}$ should be 0,

$$\begin{aligned} \rho \frac{\nabla \cdot \mathbf{u}^{n+1} - \nabla \cdot \mathbf{u}^*}{h} &= -\nabla^2 p^{n+1} \\ \rho \frac{-\nabla \cdot \mathbf{u}^*}{h} &= -\nabla^2 p^{n+1} \\ \frac{\rho}{h} \nabla \cdot \mathbf{u}^* &= \nabla^2 p^{n+1} \end{aligned} \tag{3.6}$$

we have a solution at the next time step which fulfils Equations (2.1) and (2.2). This is not the case for the intermediate velocity field $\mathbf{u}^*$. In summary, we compute the intermediate velocity field, and then Equation (3.6) must be solved for $p^{n+1}$. Having solved for this, we can calculate $\mathbf{u}^{n+1}$. If we replace $\mathbf{a}(\mathbf{u})$ by the appropriate terms from Equation (2.2), the algorithm consists in these three steps, where we calculate three quantities successively, namely $\mathbf{u}^*, p, u_{n+1}$:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{h} = -(\mathbf{u}^n \cdot \nabla)\mathbf{u}^n + \nu \nabla^2 \mathbf{u}^n + \mathbf{f}^n \tag{3.7}$$

$$\nabla^2 p^{n+1} = \frac{\rho}{h} \boldsymbol{\nabla} \cdot \mathbf{u}^* \tag{3.8}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{h}{\rho} \nabla p^{n+1} \tag{3.9}$$

The observant reader may have noticed that Equation (3.8) is a Poisson equation, which is an elliptic PDE. In other words, this step is the specific part of the algorithm where all parts of the computational domain interact at each time step. Thus the solution of Equation (3.8) accounts for the bulk of the computation time in a typical Navier-Stokes solver, so a vast amount of research has gone into developing fast solvers for this equation. For two-phase flows with large density differences, the condition number of the matrix that results when Equation (3.8) is discretised will be much larger than for the single-phase problem[147]. Since this matrix is no longer constant for the two-phase flow problem, much effort has also gone into developing fast preconditioners. The current state-of-the-art seems to be combining a (geometric or algebraic) multigrid preconditioner with a conjugate gradient method (often BiCGStab[148] ) for solving the resulting sparse linear system. In the simulations performed here,

[147] Duffy, A. *et al.* (2002)

[148] van der Vorst, H. (1992)

the BoomerAMG[149] algebraic multigrid preconditioner from the Hypre library[150] is used together with BiCGStab as implemented in the PETSc library[151] [152] to solve the linear system.

To be specific, we write the discretised Poisson problem with Neumann boundary conditions in matrix form, $Ax = b$, where $A$ is a sparse $N \times N$ matrix representing the discretised Laplacian operator, $x$ is the solution vector with $N$ elements, and $b$ is the vector holding the right-hand side of Equation (3.8). Note that here we use the language common in linear algebra texts and in numerical libraries such as PETSc, calling $x$ and $b$ vectors, even though they correspond to two- or three-dimensional data stored on the grid. Note also that $N$ is the total number of grid points, i.e. for a $200 \times 400$ grid, $N = 80000$. The matrix $A$ represents the discrete Laplacian. A good way to think of it is as coming from the convolution with a kernel $\mathbf{D}^2_{xy}$ (for the two-dimensional case). Using second-order finite differences, this kernel is given by

$$\mathbf{D}^2_{xy} = \frac{1}{\Delta^2} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \tag{3.10}$$

where the grid spacing $\Delta$ is assumed to be equal in the $x$- and $y$-directions. We note that the boundary conditions for Equation (3.8) are typically of zero Neumann type, which results in a singular matrix. This can be understood when considering $\mathbf{D}^2_{xy}$ e.g. at the left boundary $x = 0$:

$$\mathbf{D}^2_{xy} = \frac{1}{\Delta^2} \begin{bmatrix} 0 & 1 & 0 \\ 0 & -3 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \tag{3.11}$$

meaning that no value is specified at the left boundary. When this applies to all sides on the domain, no boundary conditions are given for the pressure field at all, which means the pressure is only determined up to an additive constant.

The common "engineering" approach to fixing the singularity, simply setting the pressure at some point in the domain, is not the best approach as it may pollute the spectrum of the preconditioner. Instead, projecting the discretised singular equation into the orthogonal complement of the null space of the singular matrix is a good solution[153]. To be specific, we construct a $N \times N$ projection operator

[149] Henson, V. E. *et al.* (2000)
[150] Falgout, R. *et al.* (2006)
[151] Balay, S. *et al.* (1997)
[152] Balay, S. *et al.* (2012)
[153] Zhuang, Y. *et al.* (2001)

$P$, which is applied to the right-hand side vector $b$, as

$$P = \mathbb{I} - \frac{1}{N}\mathbb{1}, \tag{3.12}$$

where $\mathbb{1}$ is the matrix having all elements equal to 1 and $\mathbb{I}$ is the identity matrix, i.e. explicitly removing the constant part of the solution. In the PETSc library that we employ here, this is achieved using the `MatSetNullSpace()` routine (along with others, see e.g. KSP Example 34 in PETSc[154] ).

Having discussed the solution of Equation (3.8), we now turn our attention to Equation (3.7), which is a time-integration step. In this equation, a forward Euler step is used. Note that in the problems we are considering, where the flow is laminar and perhaps even creeping, the diffusion of momentum represented by the second term in Equation (3.7) will dominate the convection of momentum represented by the first term.

The forward Euler method is first order in time. What can be gained by using a higher-order time integration method? As a test we employ the four-stage strong stability-preserving (SSP) Runge-Kutta (RK) method of order 3, hereafter known as SSP-RK3()4, a common choice in the literature. A key point of SSP methods is that they can handle solutions which have spatial discontinuities without introducing oscillations close to the discontinuities. We note however that the practical importance of the SSP property is debated[155]. We also note that Godunov[156] proved the following no-go theorem: a linear scheme which does not introduce new oscillations (i.e. is monotonicity preserving) can be at most first order. Our use of this time-integration scheme was inspired by the article by Kang *et al.*[157], which uses the SSP-RK3()3 method, implemented as a convex combination of Euler steps, and employed for solving the two-phase Navier-Stokes equations.

To verify that the numerical methods indeed have the claimed order, they were tested on the van der Pol equations

$$\dot{x} = y \tag{3.13}$$

$$\dot{y} = \mu(1 - x^2)y - x \tag{3.14}$$

with initial value $(0.5, 0.5)$ at $t = 0$ and the parameter $\mu = 1$. These equations were solved up to $t = 40$ for decreasing $h$ from $10^{-2}$ to $10^{-5}$, and the results were compared to the result obtained by

[154] Yang, J. (2012) [155] Ketcheson, D. I. *et al.* (2005)

using Dormand and Prince's RK 5(4)7 using a much smaller $h = 10^{-7}$. It is seen that the Euler method has order 1 and the SSP-RK3()4 method has order 3; to within machine accuracy, which is ultimately reached here for the SSP-RK3()4 method. For a readable introduction to floating-point precision and machine accuracy, the reader is referred to the first part of the classic paper by von Neumann and Goldstine[158]. Another classic which the reader may consult is the very detailed paper "What every computer scientist should know about floating point arithmetic" by Goldberg[159], who remarks that

> Floating-point arithmetic is considered an esoteric subject by many people. This is rather surprising, because floating-point is ubiquitous in computer systems.
>
> (David Goldberg)

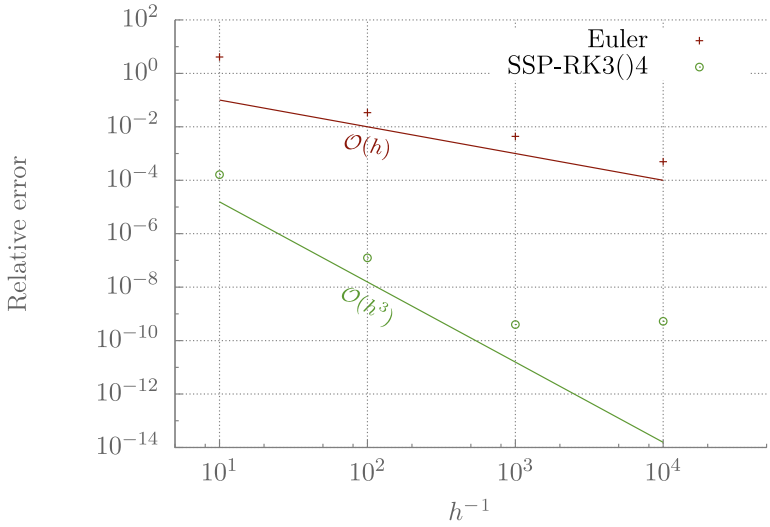



FIGURE 3.2: Order test for the van der Pol equations. It is seen that the Euler method has order 1 and that the SSP-RK3()4 method has order 3.

In order to test these methods on the Navier-Stokes equations, we need a good case to consider. Following Chorin[145], we may attempt to use the analytical 2D solution known as the Taylor-Green vortex case,

$$u = \sin x \cos y \, F(t) \tag{3.15}$$

$$v = -\cos x \sin y \, F(t) \tag{3.16}$$

$$p = \frac{\rho}{4}(\cos 2x + \sin 2x) F^2(t) \tag{3.17}$$

$$F(t) = e^{-2\nu t} \tag{3.18}$$

See also Figure 3.3 where the velocity and pressure fields are plotted for this case. Alas, using this test case does not give us much insight

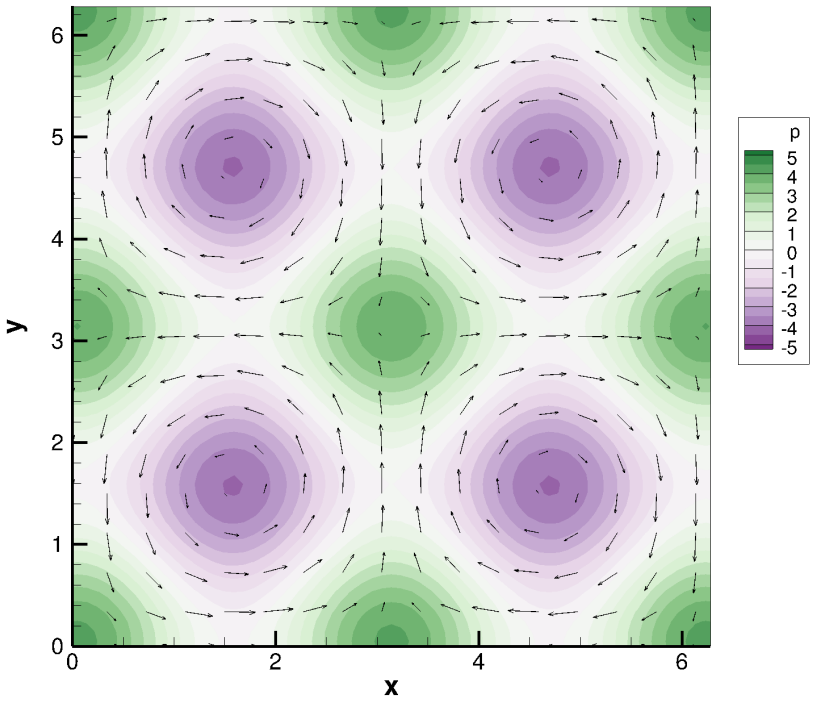



FIGURE 3.3: The Taylor-Green vortex case showing four vortices in a $2\pi \times 2\pi$ box. The velocity field is indicated with vectors and the pressure field is indicated with colours.

into the time integration method, since the spatial error dominates unless the grid size is impractically large, above 2048×2048 for the spatial discretisations used here.

We may instead use the method of manufactured solutions (MMS). This allows us to construct flow fields that have pretty much whatever properties we want, e.g. giving low error for the spatial discretisation. The construction gives an exact solution to the Navier-Stokes equations which we can use as a boundary condition and initial condition, and for computing the error at different time steps.

The construction is fairly simple. Consider $\mathbf{u} = [u\, v]$, we invent some $u = u(y,t)$ and $v = v(x,t)$ where $\frac{\partial u}{\partial x} = 0$ and $\frac{\partial v}{\partial y} = 0$, i.e. the divergence-free constraint is automatically fulfilled. Then given this $\mathbf{u}$ and some $p = p(x,y,t)$ we insert $\mathbf{u}, p$ into Equation (2.2) and get an expression for the body force $\mathbf{f}$. Using this body force in the simulations, along with the exact initial and boundary conditions, we can compare our results to the exact solution.

Following John *et al.* [160] we consider a choice for which the spatial

discretisation error will be low, namely

$$u = t^3 y^2 \tag{3.19}$$

$$v = t^2 x^2 \tag{3.20}$$

$$p = x + y \tag{3.21}$$

One may compute the body force with the computer algebra system Maple, in order to avoid calculus errors:

```
with(VectorCalculus):
SetCoordinates(cartesian[x,y]):
u := t^(3)y^(2);
v := t^(2)*x^(2);
V:= VectorField(<u,v>);
p(x,y,t) :=x+ y;
NavierStokes:= rho*(diff(V,t) +
  (V . Nabla)(V)) +
  Gradient(p(x,y,t)) -
  mu*Laplacian(V)
```

and the result may be obtained from Maple in Fortran code, to avoid errors when converting the syntax.

We consider this case starting at $t = 1.0$s. Using the Euler method and the SSP-RK3()4 method we solve the flow up to $t = 1.1$s using successively smaller time steps. We are mainly interested in problems which are viscosity dominated, so we have chosen $\rho = 1\text{kg/m}^3$, $\mu = 1\text{kg/(m s)}$ giving $\nu = 1\text{m}^2/\text{s}$. We employ a $64\times64$ grid. The largest time step used is $h = 32 \cdot 10^{-6}$ which is within the stability restriction, and we divide this time step by a factor of 2 until we reach $h = 1 \cdot 10^{-6}$. For discussion of the stability conditions used in this work, for both single- and two-phase flow, the reader is referred to Kang *et al.* [157].

The result is plotted in Figure 3.4 and is rather disappointing: we gain *absolutely nothing* by using the RK method, even though it has 4 times longer runtime. Studying the review article by Guermond *et al.* [161], this is actually as expected, as Chorin's projection method has an irreducible splitting error of $\mathcal{O}(h)$. Many authors simply employ a higher-order RK method without ever discussing the temporal error of the solution, as indeed Kang *et al.* [157] do.

One legitimate reason for using a higher order RK method is the increased stability. Another reason is that checking the actual temporal order of an implementation is very complicated for a large code with many features, and the approach we use here (method of manufactured solutions) has only recently been extended to two-phase flow. Although methods for higher-order solutions of the Navier-Stokes equations had certainly been published before the work of

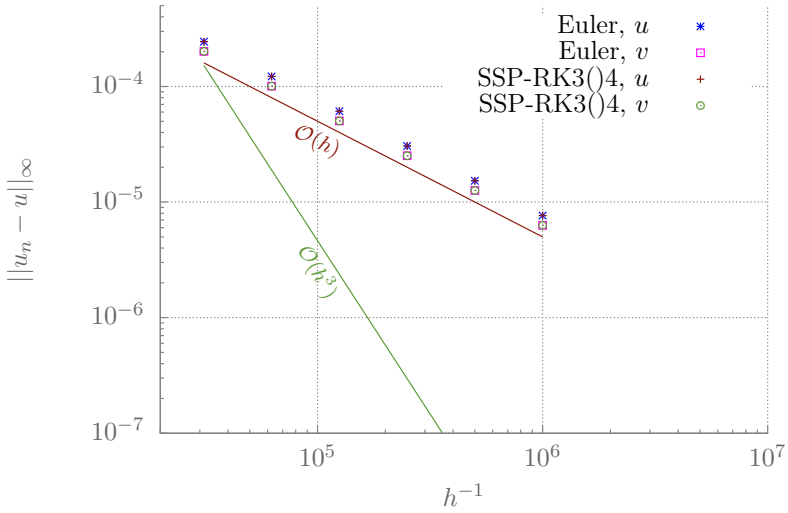



FIGURE 3.4: Order test for the Navier-Stokes equations using the method of manufactured solutions. It is seen that both methods have order 1.

Kang et al., these methods have not been adopted to a large extent in the multiphase flow community. See e.g. Pijl *et al.*[162] for a remarkable example. These authors cite the paper by van Kan[163] which popularised a second-order modification of Chorin's method, but they do not seem to employ the van Kan method and instead use the plain Chorin method.

The previously mentioned review article by Guermond et al. is highly informative with regards to order reduction for the Navier-Stokes equations. They discuss different modifications to Chorin's original scheme which allow a higher temporal order. They do, however, limit themselves to multi-step methods (BDFs in particular), but Johnston *et al.*[164] consider the case of RK methods.

One of the main topics of the article by Guermond et al. is the modification of Chorin's projection method to enable higher order. The simplest such improvement is to add $\nabla p^n$ to the right-hand side of Equation (3.7), i.e. an old value of the pressure gradient. This then transforms the Poisson equation, such that it computes not the pressure field, but the difference in the pressure field from the previous one. The boundary condition on the Poisson equation is still a zero Neumann BC. This method was popularised by van Kan[163]. See Shen[165] for a proof that this method is second order. This is called the incremental pressure projection method, and the scheme

becomes

$$\frac{\mathbf{u}^\dagger - \mathbf{u}^n}{h} = -(\mathbf{u}^n \cdot \nabla)\mathbf{u}^n + \nu\nabla^2\mathbf{u}^n + \mathbf{f}^n - \frac{\boldsymbol{\nabla} p^n}{\rho} \tag{3.22}$$

$$\nabla^2(q^{n+1}) = \frac{\rho}{h}\boldsymbol{\nabla}\cdot\mathbf{u}^\dagger \tag{3.23}$$

$$p^{n+1} = p^n + q^{n+1} \tag{3.24}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^\dagger - \frac{h}{\rho}\nabla q^{n+1} \tag{3.25}$$

Noting that $\mathbf{u}^\dagger = \mathbf{u}^* - (h/\rho)\boldsymbol{\nabla} p^n$ where $\mathbf{u}^*$ is the intermediate field in the original method, this form can be easily understood by subtracting $\nabla^2 p^n$ from both sides of Equation (3.8) in the Chorin method:

$$\nabla^2\left(p^{n+1} - p^n\right) = \frac{\rho}{h}\boldsymbol{\nabla}\cdot\mathbf{u}^* - \nabla^2 p^n \tag{3.26}$$

$$\nabla^2\left(q^{n+1}\right) = \frac{\rho}{h}\boldsymbol{\nabla}\cdot\left(\mathbf{u}^* - \frac{h}{\rho}\boldsymbol{\nabla} p^n\right) \tag{3.27}$$

$$\nabla^2\left(q^{n+1}\right) = \frac{\rho}{h}\boldsymbol{\nabla}\cdot\mathbf{u}^\dagger \tag{3.28}$$

Further extensions of this method are given in the review article by Guermond et al.. Suffice it to say here that the reason the incremental pressure form is limited to second order is that the zero Neumann BC on the pressure difference implies that the pressure gradient at the boundaries remains unchanged from the initial time. While better than the standard Chorin method, this is clearly still not entirely correct. Further improvements are made by extending this idea of redefining the variable which is solved for by the Poisson equation, so that the zero Neumann condition no longer implies unphysical conditions on the pressure field.

In the article by Kang *et al.* [157] which we follow in general here, the RK method is constructed as a convex composition of Euler steps. In each such Euler step, an update for the intermediate field $u^*$ is performed followed by a projection. Since we only care about the velocity field at the end of the time integration having $\boldsymbol{\nabla}\cdot u = 0$, this seems perhaps a bit excessive. An improvement to the method is to only perform the projection at each proper time step. This has not been considered here.

As an initial task in his MSc work, Lysgaard was suggested by the author to implement the incremental projection method, hoping that this would give a second order method. Yet again, disappointment

was to be had[166]. What was thought to be simply adding a term to the right-hand side of an equation, and some minor adjustments elsewhere in the code, turned out to be a difficult and fruitless venture. After more studies of the literature, it became apparent that no-one has published results with an *explicit* projection method of order higher than one. To the best knowledge of the author, it is an open question (that is not discussed in the literature) whether such a method can be constructed, or whether the opposite can be proven. Shen[165], Guermond *et al.*[161] and other references cited in these works all limit the discussion to implicit time stepping schemes.

## §3.2 Methods for two-phase flow

As described in Chapter 2.3 there are several additional phenomena that must be incorporated when wanting to simulate two liquids. Some of these are rather obvious: we must take into account the different densities and viscosities of the two fluids, as well as the interfacial tension, and naturally we must know where the interface is located. Let us begin by considering how to represent the location of the interface.

Many methods have been presented in the literature for this purpose. They include volume of fluid[167], level set[168], front tracking[169], phase field[170] etc. We employ here the level-set method, which is a very popular method not just for two-phase flow simulations, but also in other applications where material interfaces are important. There is a vast body of literature on the level-set method; the reader is referred to the book by Osher *et al.*[171] for an overview. At the 2013 "Short course on multiphase flow" given annually at ETH Zurich, I heard Tryggvason, of front-tracking method fame, give a remark along the lines of

> If you invent something like the level-set method, which is easy to code and has much room for being improved in various ways, a large number of people will implement and improve upon your method and cite your papers. If you invent a more complicated method that does not need so many improvements, this is not the case.
>
> (Gretar Tryggvason)

In the level-set method, the interface is represented implicitly by a scalar field that is defined on the usual simulation grid. At each point, this field $\phi$ stores the signed distance to the interface. This, in turn, means that the interface is given by the zero level set $\{\mathbf{x} | \phi(\mathbf{x}) = 0\}$,

[166] Lysgaard, M. O. (2014)
[167] Hirt, C. W. *et al.* (1981)
[168] Osher, S. *et al.* (1988)
[169] Unverdi, S. O. *et al.* (1992)
[170] Karma, A. *et al.* (1996)

which gives the method its name. All operations on the interface, such as advecting it or calculating its curvature, are performed directly on the field $\phi$. In particular, advection is done by solving the equation

$$\frac{\partial \phi}{\partial \tau} = -\mathbf{u} \cdot \nabla \phi. \tag{3.29}$$

where $\mathbf{u}$ is the flow field and $\tau$ is a pseudo-time (not the physical simulation time). An often-cited strength of the implicit formulation is that the method easily handles situations where the interface topology changes, such as the coalescence of two drops.

To visualise the level-set method, it is common to construct a $\phi$-surface embedded in three-dimensional space. This is also done here, in Figure 3.5, for the two-dimensional flow of a drop next to a flat interface (shown in gray) immersed in another fluid (shown in white) on the right-hand side of Figure 3.5). We illustrate the distance as isocontours on this figure. On the left-hand side we show the surface that corresponds to an elevation equal to $\phi$. There is a striking parallel here to a map describing an island rising out of the water. Only in this case, the "map" is reality and the "island" is the tool for describing reality.

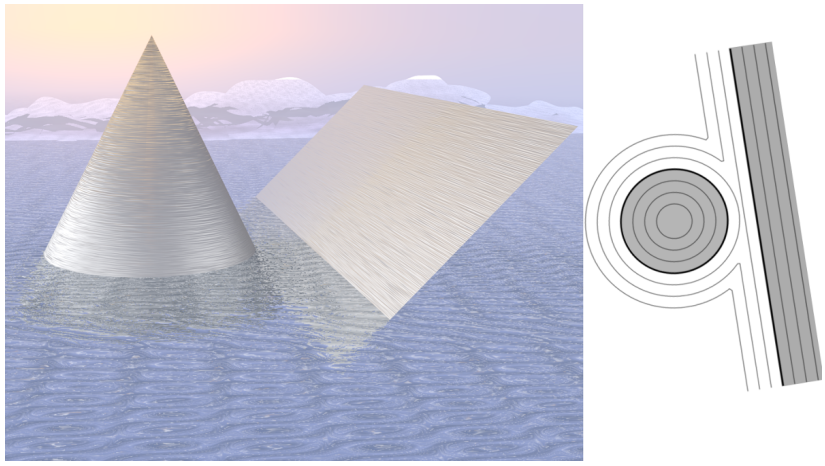


FIGURE 3.5: Illustration of the level-set method. Right: in 2D, a fluid drop (dark gray) seen next to a fluid film (dark gray), both immersed in a different fluid (white). Left: the signed-distance function representing these two fluid bodies, the drop and the film. (Figure produced by the author for Paper 2.)

Knowing the field $\phi$, and thus the location of the interface, the normal vector and curvature, $\mathbf{n}$ and $\kappa$, can be computed directly from $\phi$;

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}, \tag{3.30}$$

$$\kappa = \nabla \cdot \mathbf{n} = \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right). \tag{3.31}$$

In this context, Smereka[172] noted that

> One of the major advantages of level-set methods is their ability to easily handle topological changes. However for this problem of computing the curvature we have found this not to be the case. (Peter Smereka)

Indeed, when for instance two drops come close together, the result calculated by Equation (3.31) will be highly oscillatory, and can give rise to a numerical prevention of coalescence. A way around this problem is offered by the LOLEX method presented as part of this work[173].

One of the issues alluded to by Tryggvason in the quote above is the need for reinitialisation (or redistancing) in the level-set method. When advecting the interface according to Equation (3.29), the field $\phi$ is typically distorted such that $\phi$ is no longer a signed distance function. This is because the velocity a small distance away from the interface will be different from that at the interface. This distortion causes a loss of accuracy. To combat this, reinitialisation was introduced, whereby one periodically solves the equation

$$\frac{\partial \phi}{\partial \tau} + \text{sgn}(\phi)(|\nabla \phi| - 1) = 0. \tag{3.32}$$

This will restore $\phi$ to being a signed distance function. While a good solution, standard reinitialisation is not perfect due to numerical errors. To combat this, we further make use of the the high-ordered constrained reinitialisation method [174].

We recently pointed out[175] that the frequency at which this equation is solved should be chosen with some care; in particular for flows at low Reynolds number, where usual choices like solving it every 10 time steps will degrade the accuracy of the solution. This is because when using the stability condition (following again Kang *et al.*[157] here), the time step will be restricted by the viscous contribution to this stability condition. Neglecting the viscous contribution, one may use the stability condition to compute a convective time step, which can be orders of magnitude greater. The reinitialisation of $\phi$ is introduced to combat errors which arise due to the convection of $\phi$. It thus stands to reason that the convective time step, which is directly related to the speed at which the interface is convected (and thus deformed), is the correct time step to use as the basis for the reinitialisation frequency.

[173] Ervik, Å. *et al.* (2014)

[175] Ervik, Å. (2015)

Another popular technique to reduce the distortion of $\phi$ is that of velocity extrapolation, as introduced by Adalsteinsson *et al.*[176]. In this method, $\phi$ is advected using Equation (3.29) but with $\mathbf{u}$ replaced by a velocity field that is extrapolated from the velocity at the interface. There appears to be some confusion regarding this point in the literature; when applying velocity extrapolation, one should in theory no longer need the reinitialisation procedure. However, in practice one finds that it is still needed. The reason for this is that numerical errors are introduced when extrapolating the velocity, and reinitialisation is necessary to correct the effect of these errors.

While velocity extrapolation increases the accuracy of the solution, it is computationally costly due to the need for computing the extrapolated velocity field. Recently, Sabelnikov *et al.*[177] presented an interesting method which can be used as a substitute for velocity extrapolation at a much lower computational cost. This method has not been implemented in the present work, but appears promising for use in future work.

Having then a method for capturing the interface position, we proceed to take into account the effects of having different densities and viscosities of the two fluids, and an interfacial tension between them. Numerically, we implement this using the ghost-fluid method [178, 157], which is a sharp-interface method. This means the jumps in various quantities across the interface, as given for instance by Equations (3.34) to (3.35) below, are incorporated into the numerical stencil as jumps from one grid point to the next, as opposed to smearing out the jumps using e.g. the continuum surface force method[179]. The jump conditions at the interface read[35] [38] [180]

$$[\![\mathbf{u}]\!] = 0, \tag{3.33}$$

$$[\![p]\!] = 2[\![\eta]\!]\mathbf{n}\cdot\boldsymbol{\nabla}\mathbf{u}\cdot\mathbf{n} - \gamma\kappa, \tag{3.34}$$

$$\begin{aligned}[\![\eta\boldsymbol{\nabla}\mathbf{u}]\!] = [\![\eta]\!]\Big(&(\mathbf{n}\cdot\boldsymbol{\nabla}\mathbf{u}\cdot\mathbf{n})\mathbf{nn} + (\mathbf{n}\cdot\boldsymbol{\nabla}\mathbf{u}\cdot\mathbf{t})\mathbf{nt} \\ &- (\mathbf{n}\cdot\boldsymbol{\nabla}\mathbf{u}\cdot\mathbf{t})\mathbf{tn} + (\mathbf{t}\cdot\boldsymbol{\nabla}\mathbf{u}\cdot\mathbf{t})\mathbf{tt}\Big) \\ &- (\mathbf{t}\cdot\boldsymbol{\nabla}_\iota\gamma)\mathbf{tn},\end{aligned} \tag{3.35}$$

Here $\mathbf{n}, \mathbf{t}$ are the normal and tangent vectors at the interface, and we denote tensors formed by the outer product as e.g. $\boldsymbol{\nabla}\mathbf{u}$. We take the

[179] Brackbill, J. *et al.* (1992)
[35] Teigen, K. E. *et al.* (2009)
[38] Teigen, K. (2010)
[180] Lervåg, K. Y. (2013)

normal vector to be pointing outwards from a drop, and then the jump $[\![\cdot]\!]$ is the difference between the external and internal values, e.g. $[\![\eta]\!] = \eta_2 - \eta_1$. It should be noted that these expression have been written in a form that gives faster code when implemented, since the second and third terms in the parenthesis in Equation (3.35) contain the same product. See Lervåg[180] for a derivation and for a more thorough description of the ghost-fluid method. Note that these expressions take into account the Marangoni force created by gradients in the interfacial tension, as $\boldsymbol{\nabla}_\iota \gamma$ denotes the gradient along the interface of $\gamma$. Note also that when electric fields are applied, as in some of the work considered here, more terms and equations enter into these jump conditions. More details of this are given in Paper 4.

So far, we have assumed that the simulation domain is rectangular. This may be appropriate for some cases, such as a falling drop. In other cases, a more complex domain plays an essential role. A specific case which is important in this work is the drop being drained by a needle. In this case we want to simulate both the solid walls of the needle and the suction within the needle. To accomplish this we employ a standard $L_2$ penalisation method[181]. This method is chosen because it is very easy to implement, taking just twenty lines of code. The method can be used to simulate the effect of a solid body, and also for specifying a flow field in part of the domain, such as the needle suction. With this method, the flow field is defined in all the domain, but in the parts where the penalisation is active, the flow is forced to be approximately equal to the specified flow $\mathbf{u}_{\text{spec}}$ (which is equal to zero inside a solid body). The method consists in adding a term $(1/\eta)\chi(\mathbf{u}_{\text{spec}} - \mathbf{u})$ to the right-hand side of the Navier-Stokes equations. In this expression, $1/\eta$ is the strength of the penalisation and $\chi$ is a scalar marker function, being 1 inside a body and 0 outside it. One may think of this as a body force which is proportional to the difference between the actual flow field and the desired flow field. Proofs of the existence and uniqueness of solutions when this method is used are found in[181]. Error estimates are also given; the error is $\mathcal{O}(\eta)$. At first sight one might thus use a very small $\eta$, but the time step required for stability is proportional to $\eta$, so there is a trade off between speed and accuracy, as usual.

The author was introduced to the penalisation method at the 2014 CarteLS workshop in Bordeaux, and the method was implemented into the code employed here in one evening. The author is grateful to Chloé Mimeau for her assistance in explaining the details of the method.

[181] Angot, P. *et al.* (1999)

## §3.3 Methods for elastic interfaces

When it comes to the elastic interfaces, we are entering uncharted territory. It was far from immediately clear whether interfacial elasticity could be incorporated into the level-set/ghost-fluid method described previously. However, another method offered clear evidence of being able to implement this, namely the immersed boundary method[182] [183]. Moreover, this method has proven capable of simulating crumpling of red blood cells[184], reminiscent of the crumpled asphaltene-covered drops observed by Yeung *et al.* [141]. For these reasons, it was decided that one should try to construct a hybrid level-set/ghost-fluid/immersed-boundary method. In this work, I had invaluable assistance in M. O. Lysgaard who completed his MSc thesis[185] on this topic in 2015. The reader is referred to his thesis for the full details of the hybrid method in the two-dimensional case, as well as a good overview of the immersed boundary method and its foundations.

This method forms part of a multiscale approach, which is illustrated in Figure 3.6. The velocity field and the interface representation are stored on the Eulerian grid. The Lagrangian points from the immersed boundary method, and the tensions acting on these points, are illustrated as orange points and red/green lines, respectively. The varying color of these lines indicate the varying tensions. In the corner, a snapshot from molecular dynamics simulations indicates the use of these to estimate the properties $K_a$ and $\gamma$ of the interface. Molecular simulations are discussed in the next chapter. The proposed hybrid level-set/ghost-fluid/immersed-boundary method is validated extensively in Paper 8 and in the MSc thesis by Lysgaard[185], and the results of simulations using this method are also discussed in those works.

The immersed boundary method is concerned with Lagrangian points that represent the interface, in a break from all the methods considered so far which are based on Eulerian descriptions. In the immersed boundary method, we imagine the interface to be described by a continuum of elastic fibres, which serve only as an abstraction, having no mass and no volume. The fibres are parametrised by three coordinates, and we consider here the axisymmetric coordinate system $(\phi, r, s)$; see Figure 3.6. If one fixes two coordinates, this uniquely determines a fibre, and the remaining coordinate is a parametrisation along this fibre. For the case of an interface with no thickness, one of

[182] Peskin, C. (1977)
[183] Peskin, C. *et al.* (1995)
[184] Fai, T. G. *et al.* (2013)
[185] Lysgaard, M. O. (2015)

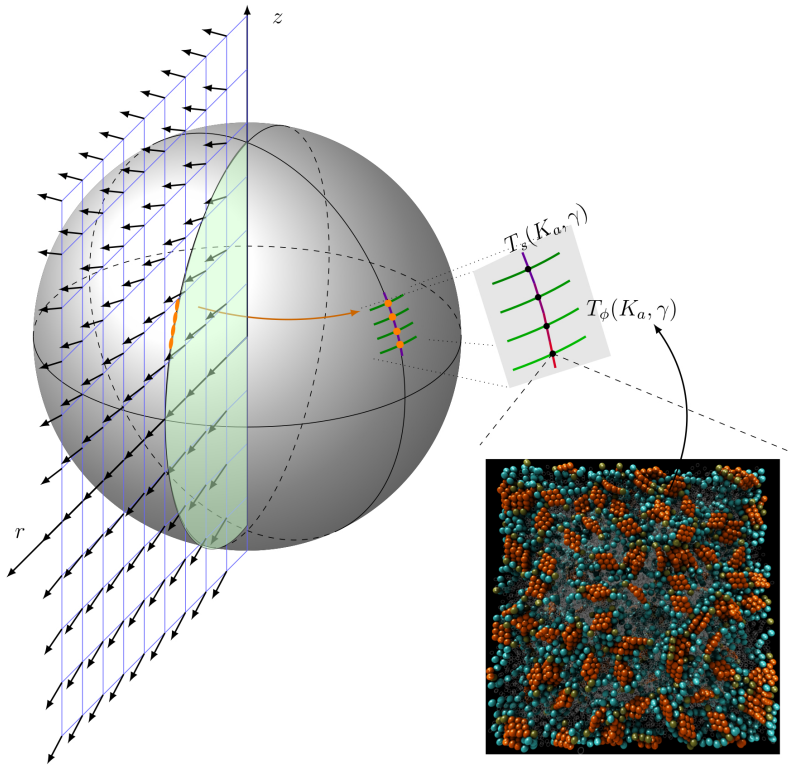



FIGURE 3.6: Illustration of the proposed method. On the Eulerian $(r,z)$ grid, the flow field $\mathbf{u}$ (vectors) and the level-set function $\varphi$ representing the green-shaded portion of the droplet are shown. Some of the Lagrangian immersed-boundary points are shown in orange, with lines indicating the tensions $T_s$ and $T_\phi$ in green and red, with varying color denoting varying tension. For a tiny patch on the interface, we compute the interfacial properties $K_a$ and $\gamma$ using molecular dynamics simulations (lower right corner). These properties are used in the calculations of the tensions. (From Paper 8.)

the coordinates is superfluous, e.g. $r = r(\phi, s)$. Furthermore, in the axisymmetric case, no properties depend on the angle $\phi$, so $r = r(s)$. Thus we are only concerned with the discretisation of a single fibre in the meridional direction on the drop.

To have an accurate representation of this single fibre, cubic splines are used. These generate a smooth analytic parametrisation, and have the further advantage that the curvature, tangent vector and normal vector are all given by analytical expressions. This is illustrated in detail in Paper 8 and by Lysgaard[185].

The tension in this fibre, in both the meridional and azimuthal directions, gives rise to forces acting on the surrounding fluids. The tensions may be just an ordinary interfacial tension, or they may originate from a more complicated expression. Note in particular that the tension in these two directions may be anisotropic.

In general, we may write down a tension $T$ at the interface, and in the two-dimensional case with both interfacial tension and elasticity, assuming a Hookean material, one obtains

$$T = K_a\left(\left|\frac{\partial \mathbf{X}}{\partial s}\right| - 1\right) + \gamma. \tag{3.36}$$

In this equation the parameter $K_a$ is the elastic modulus, and $\frac{\partial \mathbf{X}}{\partial s}$ (following the notation of Peskin) is the relative deformation of an interfacial element. It is instructive to think of the instantaneous values of $T$ as an effective interfacial tension. However there are important differences between $T$ and $\gamma$ even in the case where the interfacial tension varies e.g. due to some surfactant. Perhaps the most striking thing to note is that if $K_a > \gamma$ and $\left|\frac{\partial \mathbf{X}}{\partial s}\right|$ becomes small, this generalised tension can become zero or even negative. In the present work, we demonstrate that this is what causes the crumpled drops as observed by Yeung *et al.* [141].

Once this tension is given, a force must be computed that acts from the interfacial elements on the fluid. Following Knoche *et al.* [186] we have that the force acting in both the normal and tangential directions to the interface is given as

$$\mathbf{f} = \frac{\partial T}{\partial s}\mathbf{t} + T\left|\frac{\partial \mathbf{X}}{\partial s}\right|\kappa\mathbf{n}, \tag{3.37}$$

where we observe that the tangential force is analogous to the Marangoni force for variable interfacial tension.

Extending this to the axisymmetric case is covered in Paper 8. Following again Knoche *et al.* [186], we extended the tension to have also a component pointing in the azimuthal direction, i.e. around the symmetry axis. The two tensions are given by the symmetric expressions

$$T_s = \frac{K_a}{\left|\frac{\partial \mathbf{X}}{\partial \phi}\right|(1-\nu^2)}\left(\left|\frac{\partial \mathbf{X}}{\partial s}\right| + \nu\left|\frac{\partial \mathbf{X}}{\partial \phi}\right| - (1+\nu)\right) + \gamma, \tag{3.38}$$

$$T_\phi = \frac{K_a}{\left|\frac{\partial \mathbf{X}}{\partial s}\right|(1-\nu^2)}\left(\left|\frac{\partial \mathbf{X}}{\partial \phi}\right| + \nu\left|\frac{\partial \mathbf{X}}{\partial s}\right| - (1+\nu)\right) + \gamma, \tag{3.39}$$

where $\frac{\partial \mathbf{X}}{\partial \phi}$ represents the deformation in the azimuthal direction, which is induced by the coordinate system as the distance from the symmetry axis varies. $\nu$ here represents not the kinematic viscosity, but the Poisson ratio of the material. We consider $\nu = 1/3$ here, which is a reasonable guess for a material like asphaltenes in the absence of experimental data [187]. The force computed from these tensions is still confined to just the normal and tangential directions to the interface, since a force in the azimuthal direction would have to come

[187] Knoche, S. *et al.* (2013)

from an azimuthal derivative of the azimuthal tension $T_\phi$. Under the assumption of axisymmetry, this derivative is by definition zero, as is the azimuthal derivative of any quantity. The force is then given as

$$\mathbf{f} = \frac{\partial T_s}{\partial s}\mathbf{t} + T_s \left|\frac{\partial \mathbf{X}}{\partial s}\right| \kappa_s \mathbf{n} + T_\phi \left|\frac{\partial \mathbf{X}}{\partial \phi}\right| \kappa_\phi \mathbf{n} \tag{3.40}$$

where $\kappa_s, \kappa_\phi$ are the curvatures in the $(r, z)$ plane and in the azimuthal directions, respectively.

To compute the tensions using Equations (3.38) and (3.39) requires knowledge of the interfacial deformations. In Paper 8 we demonstrate that the level-set function, or any similar scalar marker function, does not contain the information required to compute the deformations. Equivalently, the marker function does not contain information about compression or stretching of the interface. To store information about interfacial compression or stretching in an interface capturing method which uses an Eulerian marker function, one would have to resort to additional data structures to represent interfacial strain. As we discuss in Paper 8, the hybrid level-set/ghost-fluid/immersed-boundary method is a more suitable method, since it allows the interface representation to be refined at a much lower computational cost.

The immersed-boundary method provides the required information about compression or stretching of the interface. Moreover, it has been widely used to simulate situations with a combination of elasticity and interfacial tension, such as the case of interest here. The method was originally developed for simulating the blood flow trough the heart[182], and has been a popular choice for the simulation of elastic membranes in a flow, such as red blood cells[184].

One might consider switching completely to the immersed-boundary method, but this would mean losing the sharp-interface handling of the jumps in density and viscosity. Moreover, retaining the level-set representation leaves the door open to having a changing interfacial topology in future simulations, such as for the study of drop coalescence.

Discretising the interfacial force given by Equation (3.40) is done using standard second-order central differences. Note that the Lagrangian points cannot be further apart than half the width of an Eulerian grid cell. If this happens, the discretised interfacial forces are no longer computed correctly, and the simulation becomes unstable.

[182] Peskin, C. (1977)

[184] Fai, T. G. *et al.* (2013)

The discretised forces enter the right-hand side of the discretised Navier-Stokes equations on the Eulerian grid points close to the interface. A mollified delta function is used to smear the interfacial force out to the Eulerian grid points. This delta function is the product of one-dimensional delta functions. In a break from the typical situation with smeared-interface methods, the delta function for use with the immersed-boundary method is uniquely determined by six conditions on its properties. The reader is referred to [188, sec. 6] for details of these requirements; the resulting one-dimensional delta function is given by

$$\delta(r) = \begin{cases} 0, & r \leq -2 \\ \frac{1}{8}\left(5 + 2r - \sqrt{-7 - 12r - 4r^2}\right), & -2 \leq r \leq -1 \\ \frac{1}{8}\left(3 + 2r + \sqrt{1 - 4r - 4r^2}\right), & -1 \leq r \leq -0 \\ \frac{1}{8}\left(3 - 2r + \sqrt{1 + 4r - 4r^2}\right), & 0 \leq r \leq 1 \\ \frac{1}{8}\left(5 - 2r - \sqrt{-7 + 12r - 4r^2}\right), & 1 \leq r \leq 2 \\ 0, & 2 \leq r. \end{cases} \tag{3.41}$$

where $r$ is the distance from a Lagrangian point to the Eulerian grid point which one is smearing out to.

At this point we may assemble the proposed multiscale method in its entirety. At the nanoscale, a tiny patch of the interface, of the order of 300 nm$^2$, is simulated using coarse-grained molecular dynamics. In this simulation, the (three-dimensional) simulation domain is elongated in the direction normal to the interface. From the diagonal elements of the stress tensor, the interfacial tension $\gamma$ can be computed, and by using volume-preserving deformations of the simulation domain, the elasticity $K_a$ of the interface can be estimated. The complex effect of asphaltene molecules at the interface is considered, using several different models based on a group-contribution approach. Details of the molecular simulations are given in the next chapter.

At the macroscale, an axisymmetric drop is simulated using a combination of Eulerian and Lagrangian descriptions. The flow is governed by the incompressible Navier-Stokes equations, which are solved on a structured grid using finite differences, a Runge-Kutta time integration and the Chorin projection method. The density and viscosity differences are handled using a standard level-set/ghost-fluid approach, where e.g. the jump in the pressure across the interface is enforced in a sharp manner. The generalised tension on the interface, resulting from the combination of interfacial tension and elasticity, is

implemented using the immersed-boundary method. The Lagrangian points which discretise the interface in the immersed-boundary method give the interfacial deformations required to compute the tensions. The force resulting from the tensions is distributed to the Eulerian grid and enters into the discretised Navier-Stokes equations. The level-set function, stored at the Eulerian grid points, is computed from the Lagrangian interface representation in each time step. In summary, we have at the macroscale a hybrid level-set/ghost-fluid/immersed-boundary method having both sharp and smooth interface discretisations.

## §3.4 Methods for molecular simulation

Molecular dynamics is, on the face of it, a very simple method. $N$ particles are placed in a virtual box, typically with periodic boundary conditions. Commonly, these particles are taken to be atoms, and the forces between atoms are taken to be fully classical forces (though they are quantum-mechanical in origin). The force field which is prescribed gives the acceleration on each particle as a function of the locations of all the particles. Then one simply does numerical integration of Newton's second law of motion for all the particles, and the system evolves in time. If we denote the position of each particle by $\mathbf{x}_i$, with mass $m_i$, the equations of motion are

$$\frac{\partial^2 \mathbf{x}_i}{\partial t^2} = \sum_{j \neq i} \frac{F(|\mathbf{x}_j - \mathbf{x}_i|)}{m_i} \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|} \tag{3.42}$$

where we assume that the forces are conservative, and that the forces between two particles only depend on the distance $r$ between them, such that $F(r) = -\partial V(r)/\partial r$. One must then specify the potential $V(r)$. The force field is short-ranged, typically decaying like $r^{-6}$, so a cut-off is employed such that only the closest neighbours contribute to the force on a particle.

To proceed, since only the two-body problem has an analytical solution, these equations of motion must be integrated numerically. A symplectic integration method is typically used, to ensure that the energy drift is very small and that simulations are stable over very long times. A popular example is the velocity Verlet method, which given the velocities $v_i^n$ and positions $x_i^n$ at the time step $n$ computes

the positions and velocities at the next time step $n+1$ as

$$\mathbf{v}_i^{n+1/2} = \mathbf{v}_i^n + \sum_{j\in\mathbb{N}(i,n)} \frac{\Delta t}{2m} F(|\mathbf{x}_j^n - \mathbf{x}_i^n|) \frac{\mathbf{x}_j^n - \mathbf{x}_i^n}{|\mathbf{x}_j^n - \mathbf{x}_i^n|} \tag{3.43}$$

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \mathbf{v}_i^{n+1/2}\Delta t \tag{3.44}$$

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^{n+1/2} + \sum_{j\in\mathbb{N}(i,n+1)} \frac{\Delta t}{2m} F(|\mathbf{x}_j^{n+1} - \mathbf{x}_i^{n+1}|) \frac{\mathbf{x}_j^{n+1} - \mathbf{x}_i^{n+1}}{|\mathbf{x}_j^{n+1} - \mathbf{x}_i^n|} \tag{3.45}$$

Here $\Delta t$ is the time step, and $\mathbb{N}(i,n)$ is the neighbour-list, i.e. the list of all particles which are within the cutoff distance $r_{\text{cut}}$ of particle $i$ at time step $n$. $r_{\text{cut}}$ is chosen such that the potential is approximately zero for $r > r_{\text{cut}}$. The use of a neighbour-list dramatically speeds up molecular dynamics simulations, at the cost of increased memory use.

First developed around 1955, molecular dynamics was found (in spite of predictions made in the statistical mechanics community) to be a very powerful method even with the computers at the time. Molecular dynamics results presented by Alder *et al.*[189] gave the world an unprecedented view of the microscopic world, and helped solve important open problems such as the hypothesised existence of a solid-liquid phase transition in a system of hard spheres[190]. In Figure 3.7, Berni Alder is seen standing behind Mary Ann Mansigh and Tom Wainwright, studying some of the first images produced by molecular dynamics simulations. Alder and Wainwright wrote an early introduction to molecular dynamics[191]. Good textbooks on the subject include[192][193]. Care has to be taken when programming a molecular dynamics code; both in order to obtain the correct dynamics (such as energy conservation) and to have a fast implementation. The reader is referred to the mentioned textbooks for details.

It is noteworthy (and sad) that the early female programmers, like Mary Ann Mansigh who wrote the first molecular dynamics code, or Mary Tsingou who wrote the code for the Fermi-Pasta-Ulam model in 1954, were not included as authors on the papers describing the results.

In the 60 years after its introduction, molecular dynamics has undoubtedly come of age. Simulations have been performed with hundreds of billions of atoms[194]. Others have simulated entire HIV virus capsule structures[195]. Yet there remains a fundamental speed limit to how fast fully atomistic simulations can go. Once the number of atoms per CPU core becomes too small, strong scaling of the system levels out, and throwing more cores at the problem does not reduce the time-to-solution.

[190] Alder, B. J. *et al.* (1957)
[191] Alder, B. J. *et al.* (1959)
[192] Allen, M. P. *et al.* (1989)
[193] Frenkel, D. *et al.* (2001)
[194] Kadau, K. *et al.* (2006)
[195] Zhao, G. *et al.* (2013)

FIGURE 3.7: Berni Alder stands behind Mary Ann Mansigh (who wrote the first molecular dynamics code) and Tom Wainwright. They are studying oscillograph traces from the simulations. (Image courtesy of the AIP Emilio Segre Visual Archives.)

One way around this conundrum is to employ coarse-grained molecular dynamics. Here, the particles being simulated are no longer atoms, but "beads" representing two to four atoms (not counting hydrogen). This has a multitude of advantages: the total number of particles is reduced. Sticky interactions such as hydrogen bonding that may keep the system in a local quasi-equilibrium state are averaged out. And electrostatic effects, which are long ranged and thus costly to simulate, also disappear. All in all, studies in the literature have reported speedups of three orders of magnitude or more. Combined with the recent development of GPU hardware to accelerate the simulations, coarse-grained simulations can probe complex systems, such as the mixture of oil with asphaltenes and water, at temporal and spatial scales that are (and will likely remain) unavailable to studies using atomistically detailed approaches.

Several coarse-graining procedures exist in the literature. It is common to distinguish between bottom-up approaches, that attempt to tune the coarse-grained models to atomistic ones, and top-down approaches, that start with thermophysical properties of the fluids in question and build models based on those. See e.g. Brini *et al.*[196] for a review of coarse-graining methods. The particular method employed here, namely the SAFT-$\gamma$ Mie approach[197], is of the latter type. SAFT stands for Statistical Associating Fluid Theory, $\gamma$ indicates that

[197] Müller, E. A. *et al.* (2014)

this is the third "version" of the approach, and Mie refers to the Mie potential[198] that is employed, *viz.*

$$V(r) = C(n,m)\,\epsilon\left[\left(\frac{\sigma}{r}\right)^n - \left(\frac{\sigma}{r}\right)^m\right], \tag{3.46}$$
$$C(n,m) = \left(\frac{n}{n-m}\right)\left(\frac{n}{m}\right)^{m/(n-m)}.$$

This potential strongly resembles the Lennard-Jones potential discussed previously, except that the two exponents are no longer fixed to the values 12 and 6. The normalisation factor $C(n,m)$ ensures that the well depth remains equal to $\epsilon$ for any $n, m$. In the $n \to \infty$ limit the "width" of the potential well goes to zero. Note also that the limit $\lim_{n\to m} C(n,m)$ does not exist, but the one-sided limits do exist: $\lim_{n\searrow m} = +\infty$ and $\lim_{n\nearrow m} = -\infty$.

It may be verified by simple algebra that the Mie potential is *invariant* under the interchange of $n$ and $m$, since both the prefactor $C(n,m)$ and the term in the square parenthesis in Equation (3.46) are antisymmetric under this interchange.

Since it has been shown[199] that the two exponents $n$ and $m$ are not independent, but rather intimately related, we fix $m = 6$ and let $n$ vary. The effect of varying $n$ is illustrated in Figure 3.8, where we plot the potential with $n$ going from 8 to 24 in steps of 2, for $\epsilon = 200$, normalising $r$ by $\sigma$. Alternatively, we may plot the potential with $r$ normalised by the location of the potential minimum, $r_0$. This is shown in Figure 3.9. One easily obtains $r_0$ as a function of $\sigma$ and $n$ by finding the zero of the derivative of the potential, *viz.*

$$r_0 = \sigma\left(\frac{n}{m}\right)^{\frac{1}{n-m}}. \tag{3.47}$$

From these figures is seen that a lower $n$ corresponds to a more long-ranged or "softer" potential, while in the limit $n \to \infty$ the potential becomes equal to that of a hard sphere with radius $\sigma$. The $n = 8$ potential is the softest considered in this work, used in the models of water created by Lobanova *et al.*[200].

In the SAFT-$\gamma$ Mie approach, there is a one-to-one correspondence between the parameters of the SAFT Equation of State (EoS) and those of the intermolecular potential used in the simulations. By fitting the EoS to thermophysical properties, e.g. the vapor and liquid densities as a function of temperature, one fixes the parameters of the EoS and the potential. The key point of the approach is that molecular dynamics simulations using the potential with these parameters give the same thermophysical properties as the EoS and the experimental data used to fit the EoS. Furthermore, when simulations are performed to predict properties not used in the original fit, the agreement with

[198] Mie, G. (1903)

[199] Ramrattan, N. *et al.* (2015)

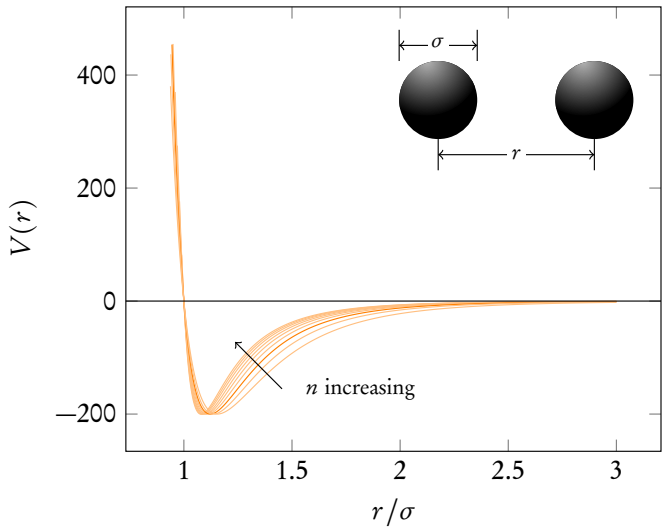



FIGURE 3.8: The Mie (n,6) potential shown for $n$ varying from 8 to 24. The Lennard Jones (12,6) potential is shown in a stronger colour.

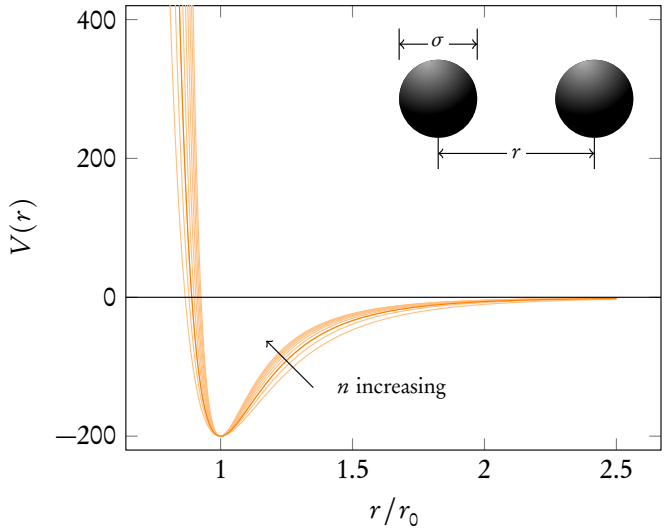



FIGURE 3.9: Same as the previous figure, but normalising the radial coordinate by the potential minimum $r_0$ instead of $\sigma$.

experimental data is very good. There is a large body of work in the literature that has used this approach to study a wide range of simple as well as complex fluid systems [201] [202] [203] [204] [205] [206] [200] [207] [208]. Also noteworthy is the correlation presented by Mejía *et al.* [209], which instead of fitting the EoS to experimental data uses the critical properties of the fluid to compute the force field parameters. This is significant, since running the correlation is much quicker and requires less input data from experiments.

[201] Avendaño, C. *et al.* (2011)
[202] Avendaño, C. *et al.* (2013)
[203] Lafitte, T. *et al.* (2012)
[204] Müller, E. A. *et al.* (2014)
[205] Herdes, C. *et al.* (2015)
[206] Herdes, C. *et al.* (2015)
[200] Lobanova, O. *et al.* (2015)
[207] Lobanova, O. *et al.* (2016)
[208] Theodorakis, P. E. *et al.* (2015)

The link between a thermodynamic equation of state and an intermolecular potential goes back to the van der Waals equation[210]. This may serve as an illustration of the link between the Mie potential and the equation of state for the SAFT-$\gamma$ Mie approach. The van der Waals equation of state is

$$P = \frac{RT\rho}{1-b\rho} - a\rho^2 \tag{3.48}$$

where $P$ is the pressure and $\rho$ is the density, $R$ is the universal gas constant and $T$ is the temperature, and finally there are two parameters $a$ and $b$. The equation of state is fitted to experimental data for some fluid by determining values for $a$ and $b$. It can be shown that this equation of state is linked to a Sutherland pair potential,

$$V(r) = -\epsilon\left(\frac{\sigma}{r}\right)^6 \tag{3.49}$$

and the two parameters $\epsilon$ and $\sigma$ here are directly related to the parameters $a$ and $b$ by the formulae

$$a = \frac{2}{3}N_{\mathrm{Av}}^2\pi\epsilon\sigma^3 \tag{3.50}$$

$$b = \frac{2}{3}N_{\mathrm{Av}}\pi\sigma^3 \tag{3.51}$$

where $N_{\mathrm{Av}}$ is Avogadro's number. In the same way, an explicit link is obtained between the equation of state and the Mie potential in the SAFT-$\gamma$ Mie approach; then there are four parameters and the underlying equations are more complicated, so the expressions are more involved. The reader is referred to Mejía *et al.*[209] for the full details.

As part of the present work, two computational tools have been developed for working with the SAFT-$\gamma$ Mie approach, presented in Paper 6 and Paper 7. These are discussed in further detail in Chapter 4.2.

The Mie potential in this approach gives the force between two identical coarse-grained beads. We will consider now first molecular models which are *homonuclear*, i.e. that a model for some molecule is made up of several beads which are tangentially bonded together in a linear shape, like pearls on a necklace.

When a mixture of different compounds is considered, a *mixing rule* must be used to obtain the potential (and thus the force) which acts

[210] van der Waals, J. D. (1873)

between beads of different types. With the SAFT-$\gamma$ Mie approach it is common to use the Lafitte mixing rules[202]. These rules, given below, specify the parameters marked by $^\dagger$ for the Mie potential between beads of two different types $i$ and $j$.

$$\sigma^\dagger = \frac{\sigma_i + \sigma_j}{2} \tag{3.52}$$

$$n^\dagger = 3 + \sqrt{(n_i - 3)(n_j - 3)} \tag{3.53}$$

$$\epsilon^\dagger = (1.0 - k_{ij}) \left( \frac{\sqrt{\sigma_i \sigma_j}}{\sigma_\dagger} \right)^3 \sqrt{\epsilon_i \epsilon_j} \tag{3.54}$$

In these mixing rules there is one free parameter $k_{ij}$. This parameter may be tuned to obtain the correct interaction between two beads of different type, by comparing computed results for some physical property such as interfacial tension or solubility to experimental values.

To extend this to more complicated molecules, such as surfactants and asphaltenes, it will no longer be sufficient to have molecular models where all beads are of the same type. This means we will require a *heteronuclear* model, where several different beads are tangentially bonded together in the same molecule. To construct these models, a group-contribution approach is used. This means that one constructs the models for individual beads based on the thermophysical properties of that that subgroup of the molecule, as if it were a separate molecule.

For instance, if one wants to model octanol, an alcohol with a long tail, we might construct one bead to represent the part containing the hydroxyl group and two methyl groups. This part of the molecule corresponds to ethanol, so we would use ethanol's physical properties to obtain the force field parameters for this bead. The remaining part of the molecule is a linear alkane, namely hexane, and we would use the properties of hexane to construct the force field parameters for this. Finally, one must put these beads together in a combination which resembles the original molecule. In this case, using one ethanol bead and two hexane beads together in a linear chain gives a good correspondence. This is illustrated in Figure 3.10, where the blue circle represents the ethanol bead and the green circles represent the hexane beads.

When considering a mixture of fluids which are immiscible, such as oil and water, we are typically interested in computing the interfacial

[202] Avendaño, C. *et al.* (2013)

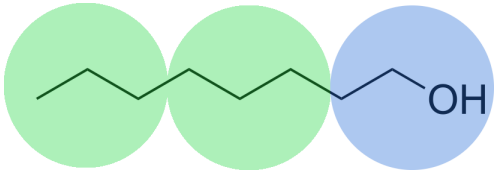



FIGURE 3.10: Illustration of how one might construct a coarse-grained model for octanol using the group-contribution approach.

tension. To achieve this, it is necessary to have well-defined and flat interfaces in the simulation box. This is achieved by using an elongated box, $L_x = L_y < L_z$, where the tendency for interfaces to attain the minimum energy configuration ensures that the interfaces will be perpendicular to the elongated direction. Since periodic boundary conditions are used, there must always be an even number of interfaces in the system.

Assuming the system has two interfaces, one may compute the interfacial tension from the diagonal elements of the stress tensor, $\sigma_{ii}$, as

$$\gamma = \frac{1}{2}\int_0^{L_z}\left(-\sigma_{zz} + \frac{1}{2}(\sigma_{xx} + \sigma_{yy})\right)dz. \tag{3.55}$$

This is known as the mechanical route for computing the interfacial tension, and was introduced by Kirkwood *et al.* [211]. Since this integral may be split into three parts, it is typical for molecular dynamics software to compute the "pressure tensor" elements $p_{xx}$ (and similarly for the $y$ and $z$ directions) as

$$p_{xx} = \int -\sigma_{xx}dx, \tag{3.56}$$

From these quantities we may compute $\gamma$ using Equation (3.55). The value of $\gamma$ computed at a single instant in time using this method will fluctuate significantly from one time step to the next. Thus time averages are employed to obtain a reliable estimate of $\gamma$.

The elasticity of the interface may be computed as the change in $\gamma$ when the interfacial area is changed in such a way that the volume of the simulation domain remains constant. This means changing the box dimensions from their original values $L_x, L_y, L_z$ to

new values

$$L'_x = \frac{L_x}{\sqrt{a}} \tag{3.57}$$

$$L'_y = \frac{L_y}{\sqrt{a}} \tag{3.58}$$

$$L'_z = a\,L_z \tag{3.59}$$

where $a$ is the desired scaling factor. Following den Otter *et al.*[212], Boek *et al.*[213], denoting the change in tension by $\Delta\gamma$ and changing the interfacial area from $A_0$ to $A$, the elasticity is given by

$$K_a = \frac{\Delta\gamma}{A/A_0 - 1}. \tag{3.60}$$

First discussed by Lord Kelvin in 1874[214], popularised by Johann Loschmidt in his critique of Boltzmann's work proposing that the entropy is non-decreasing.

The fidelity of molecular dynamics as a simulation tool was never in question, but many (including Kirkwood and Edward Teller) doubted it would be fast enough to be competitive with Monte Carlo methods.

As a concluding remark, it is noteworthy how molecular dynamics conjures up that age-old conundrum of statistical physics, namely Loschmidt's paradox (previously discussed in Chapter 2.1). This is the question of how the second law of thermodynamics (which is time-irreversible) can arise from time-reversible dynamics. How do our simulations of time-reversible equations of motion, using numerical methods that are also time-reversible, give rise to irreversible processes such as the aggregation of asphaltenes and their adsorption at the interface? This is intimately related to questions of ergodicity. The unprovenness of the ergodic hypothesis, i.e. that time-averages and ensemble-averages give the same answers, caused a great deal of uncertainty in the early development of the Monte Carlo method. John Kirkwood was reportedly among those expressing doubt[215]. The relief was great when his former student, Berni Alder, demonstrated that molecular dynamics and Monte Carlo simulations gave the same answers. Today one has great faith in the ergodic hypothesis being true, yet a rigorous proof remains elusive. One should also note that there is a concept of *ergodicity breaking* which occurs in macroscopic physical systems, giving rise e.g. to the observable magnetisation in ferromagnetic materials.

Coming back to the question of how time-irreversibility arises, statistical physicists speak of this in the language of symmetry breaking. For systems like molecular dynamics, where one speaks of *exploring the phase space*, recent work by Hoover et al.[216][217][218] on model non-equilibrium systems provides a fascinating insight into this.

[215] Battimelli, G. *et al.* (1990) [217] Hoover, W. G. *et al.* (2015) [218] Hoover, W. G. (2015)
[216] Patra, P. K. *et al.* (2015)

In these systems, the phase space contains two emergent structures called an attractor and a repellor. Such structures are a recurring theme in nonlinear dynamics (colloquially known as chaos theory); see e.g. the book by Strogatz[219] for a good introduction.

When time proceeds in the normal direction, the point in phase space corresponding to the system state "flows" from the repellor towards the attractor. When time is reversed, the flow goes the other way. The interesting thing is that for the model non-equilibrium systems considered by Hoover et al.[216][217][218], it is found that these two structures in phase space are qualitatively different: the repellor is a fractal structure with measure zero, while the attractor is a fractal structure with measure one. This "volumetric" difference causes the symmetry breaking in a very neat fashion. While this result is obtained for systems with a very low-dimensional phase space as compared to the typical case in molecular dynamics (where the dimension of phase space becomes of the order of millions), the explanation holds a certain elegance that makes it attractive as an explanation of the behaviour observed in molecular dynamics.

The *measure* is a mathematical concept that may be foreign to the reader. If we imagine the fractal as some three-dimensional structure, for instance like the branches of a bush, the measure is analogous to the volume of the branches.

CHAPTER 4

# Computational tools

Contents


4.1 Reproducibility and open science . . . . . 63
4.2 Tools developed in the present work . . . . 64
4.3 Tools used in the present work . . . . . 66


An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship.

David Donoho

## §4.1 Reproducibility and open science

Reproducibility has become a hot topic lately, not just in the experimental sciences and the social sciences, but also for computational research. The concept of open source software is widespread in many fields, and major scientific codes are open-sourced both in the field of molecular dynamics and in the field of computational fluid dynamics. The idea of reproducible computational research traces back to Buckheit and Donoho[220] and their influential 1995 book about the WaveLab software and reproducible research, and to Jon Claerbout who championed the concept in geophysics in the early 1990's. The latter stated[221]

> I discovered reproducibility in computational research when I learned about makefile syntax and how to use it to incorporate figures in documents. (Jon Claerbout)

Both Claerbout and Donoho lamented that their ideas were not taken up into the mainstream. Today, we are hopefully on the road to getting there.

One improvement that has helped this cause immensely is modern distributed version control systems such as `git`[222]. When you update your code today, perhaps with a new model or to fix a bug, and this code is hosted at a public repository like Github or Bitbucket, anyone

So it only took 20 years for the idea of reproducible computational research to catch on. At least we are in a better state than psychology, where the recently "discovered" reproducibility crisis has been apparent to the keen observer at least since Feynman's "cargo cult science" speech in 1974.

[220] Buckheit, J. B. *et al.* (1995)

[222] Torvalds, L. *et al.* (2005)

in the world can obtain the update quickly and effortlessly. This has inspired the concept of *social coding*, where workflows are now available that let anyone submit *pull requests* with fixes or new features to your code. You and your team can then review the pull request, comment on it, and (potentially after a few rounds of correction) you can merge the request into your code. The reader has probably noticed the strong parallels with the peer review process in science. Such workflows have been or are being adopted by many, if not most, of the widely used open-source scientific codes and libraries available today. The tools developed in the present work have also adopted this open-source and git-based model.

## §4.2 Tools developed in the present work

As stated previously, two computational tools have been developed as part of the present work. These tools work together to give an unprecedented ease-of-use of coarse-grained molecular dynamics simulations, from the process of obtaining force field parameters representing the fluid in question, to the point where the simulation is running on your computer. The tools have been named raaSAFT (Paper 6) and Bottled SAFT (Paper 7), both names being wordplay on the fact that SAFT means "fruit juice" in the author's native tongue.

Let us begin by discussing Bottled SAFT (Paper 7). This tool originated from the observation that one should be able to combine the correlation given by Mejía *et al.* [209], colloquially known as the M&M correlation, with the large tables of experimentally measured critical properties that exist in the literature. In building Bottled SAFT, we have taken the book by Yaws [223], containing critical properties of more than 7800 fluids, and applied the M&M correlation to this large data set. By employing the excellent Pandas library [224] [225], the data could be managed in an automated fashion. The data set was filtered to exclude results that were outside the range of application of the correlation. In the end, results for more than 6000 fluids were obtained. Using the equally excellent Flask framework [226], a web application was constructed where users may search the database for the desired fluid, by inputting the CAS number, chemical formula, or name. The force field parameters are then given as a table. They are also given as a script implementing this particular fluid into raaSAFT, which we will discuss next. This script can be saved to one's computer

[224] McKinney, W. (2010) [225] McKinney, W. (2015) [226] Ronacher, A. (2015)

and the model used directly in molecular dynamics simulations. The website is fully operational at `http://www.bottledsaft.org`, and the code implementing the website as well as the database of force field parameters is freely available at Bitbucket (`http://www.bitbucket.org/asmunder/bottledsaft`).

raaSAFT (Paper 6) is the other tool built during this thesis. It, too, is fully open source and available at `http://www.bitbucket.org/asmunder/raasaft`. Being a Python code, it can be installed on any computer with the single command `pip install raasaft` thanks to distribution on PyPI[227]. It does, however, depend on another piece of software that takes slightly more effort to install, namely HOOMD-blue[228] [229]. HOOMD-blue is a modern molecular dynamics code written to run on GPUs, which is fast and highly scalable. It is also easily scriptable since the user interface takes the form of Python scripts. HOOMD-blue does not implement any particular force fields, but can be used to simulate any kind of atomistic or coarse-grained molecular dynamics, as well as other particle-based methods like dissipative particle dynamics. The purpose of raaSAFT, then, is to implement the SAFT-$\gamma$ Mie approach on top of HOOMD-blue. By leveraging the best features of Python, a language that borrows both from object-oriented and from functional programming paradigms, raaSAFT makes the implementation of different simple or complex fluids a matter of few lines with no repetitive coding. Implementing the wide range of models published with the SAFT-$\gamma$ Mie approach, raaSAFT makes it easy for anyone to replicate the results in these works.

A noteworthy feature of raaSAFT is that it explicitly encourages users to send in jobscripts used to run simulations for their publications, such that these simulations are easily reproducible by others. A dedicated folder in the git repository holds reproduction jobscripts for various papers. For the molecular simulations considered here in Papers 5, 6 and 8, jobscripts will be uploaded to this folder when the papers are published.

To illustrate how simple the syntax is for adding models even for complex molecules, we reproduce on the next page a code listing from Paper 6 that contains the implementation of a polystyrene model. This model can be used in user jobscripts by calling the constructor at a single line in the jobscript. Note that both the number of polystyrene molecules and the number of monomers making up each molecule are given as arguments to the constructor. This lets the user easily perform

[227] Jones, R. (2002)
[228] Anderson, J. A. *et al.* (2008)
[229] Glaser, J. *et al.* (2015)

parameter studies e.g. of the effect of polystyrene concentration and size for a given solvent system. It also makes it very simple to study polydisperse systems containing polymer molecules of different size, to study different solvents or mixtures of solvents, etc.

```
1  class Polystyrene(MieCGHet):
2      def __init__(self,count,monomers):
3          self.BBN = BackbonePS(count=1)
4          self.BCH = BenzenePS(count=1)
5          self.Components=[self.BBN, self.BCH]
6          self.Name = "Polystyrene"
7          self.ShortName = "PS"
8          self.Segments = 2*monomers
9          self.Def = ([self.BBN]+[self.BCH])*monomers
10         MieCGHet.__init__(self,count)
11         self.NrBondTypes = 2
12         bt1, bt2, = self.getBondNames()
13         self.Bonds = []
14         for m in range(monomers-1):
15             self.Bonds.extend( [ (2*m,2*m+1,bt1), (2*m,2*m+2,bt2) ] )
16         self.Bonds.append((2*(m+1),2*(m+1)+1,bt1))
17         self.createBondSpec([10000,6309.5])
18         self.Citation = "DOI: unpublished"
```

To briefly explain this script: the first two lines specify the building blocks of the molecule, an aliphatic and an aromatic bead. Together these form a styrene monomer. They are implemented by two other classes, each being eight lines of code specifying the Mie parameters, molar mass etc. In line 5 these two building blocks are put into the list of components forming the polystyrene model. The model is then given a name, a short name, and the number of beads (or segments) is specified. The `self.Def` structure in line 9 numbers the beads in each molecule, here in an alternating sequence of aliphatic and aromatic beads. Lines 11-17 specify the bonds, of which there are two types in this model: aliphatic-aliphatic bonds forming the molecule backbone, and aliphatic-aromatic bonds connecting the aromatic beads hanging off the backbone. These bonds are specified in the `for` loop in lines 14-15, except for the final bond which is set in line 16 (since this bead has no bond to the next aromatic bead). Finally, the two bond constants are set, along with the DOI (this model is not yet published).

## §4.3 Tools used in the present work

In addition to raaSAFT and Bottled SAFT, many computational tools have been used in this work. The previously mentioned HOOMD-blue molecular dynamics code [228] is one of these, which is used

together with raaSAFT for all the molecular simulations performed as part of this work.

Other tools include the Tecplot[230] and VisIt[231] scientific visualisation softwares, as well as the general-purpose open-source 3D graphics software Blender[232]. The latter is a particularly interesting tool since it is primarily built for movie and game designers. This means it can be used to achieve photorealistic renderings both for illustrations (cf. Figures 1.1 and 3.5) and for rendering representations of fluid interfaces which have been produced from simulations, as in Figures 4.1 and 4.2 shown in the margin (reproduced from Paper 4 and Paper 8, respectively).

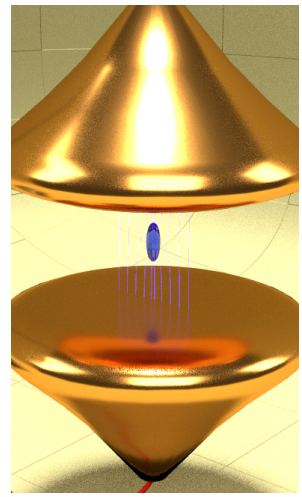


FIGURE 4.1: A drop is stretched by the electric field caused by a voltage applied across two electrodes. (From Paper 4.)

FIGURE 4.2: An asphaltene-covered drop is deflated by a needle, causing the interface to become tensionless and crumple near the neck. (From Paper 8.)

For creating line plots in a self-documenting and reproducible fashion, RMarkdown has been the tool of choice. This combination of the statistical programming language R[233], which has the excellent plotting library ggplot[234], with Markdown documents where one can write mathematical symbols and equations using LaTeX and cite the scientific literature using DOI strings, is very powerful. An RMarkdown document typically takes input data, performs calculations on the data, and plots or tabulates the results, all intertwined with text that describes the data. This is an example of *literate programming*, a paradigm introduced in the eponymous paper by Knuth[235]. Also noteworthy in RMarkdown is that the text can contain expressions computed directly from the data, e.g. the code `$\xi$ = ‘r xi_val‘` renders as $\xi = 0.245$ where “0.245” is inserted as the result of some R computation, e.g. a parameter obtained by fitting an expression to experimental data.

Finally, a tool which has been very important in this work is the in-house two-phase incompressible Navier-Stokes solver called “levelZ”. This code uses the previously discussed numerical methods such as the Chorin projection method, the level-set and ghost-fluid methods, etc. Development started in 2006, and the code is written in modern Fortran, making use of the PETSc[151] library for solving the pressure Poisson equation, as discussed in Chapter 3.1.

Results from the code have been used in many publications concerning two-phase flow phenomena[35][36][236][237][238][173][239][240][241][242]. The code can be used to simulate two-phase flow in two

[230] The Tecplot Team (2016)
[231] Childs, H. *et al.* (2005)
[232] Blender Community (2015)
[233] R Core Team (2013)
[234] Wickham, H. (2009)
[151] Balay, S. *et al.* (1997)
[35] Teigen, K. E. *et al.* (2009)
[36] Teigen, K. E. *et al.* (2010)
[236] Teigen, K. E. *et al.* (2010)
[237] Lervåg, K. Y. *et al.* (2013)
[238] Brunsvold, A. L. *et al.* (2013)
[173] Ervik, Å. *et al.* (2014)
[239] Ervik, Å. *et al.* (2014)
[240] Gjennestad, M. *et al.* (2015)
[241] Ervik, Å. *et al.* (2016)
[242] Ervik, Å. *et al.* (2016)

dimensions and in axisymmetric configurations. The hybrid level-set/ghost-fluid/immersed-boundary method developed for Paper 8 is implemented in this code. From previous work by Teigen *et al.* [35] [36] [236], the code can be used to simulate the effects of an electric field on drops, as well as the effect of insoluble surfactants at the drop interface. These capabilities are used in Papers 3 and 4.

Paper 2 reports on the initial work to parallelise a three-dimensional version of the code. This lays the foundations for future 3D parallel simulations of electrocoalescence and other two-phase flow phenomena. After Paper 2 was presented at the CFD 2014 conference, some additional work was done to test the code, which is reported here. In particular, the code was tested on the standard three-dimensional Taylor-Green vortex test case. This is a Direct Numerical Simulation (DNS) of a single-phase flow that develops turbulent structures. These simulations were done on the vsl176 cluster at SINTEF Energy Research, and later also on the Vilje cluster at NTNU. The specific case considered was the Taylor-Green vortex with parameters given in Test Case C3.3 of the 3rd International Workshop on High-Order CFD Methods. The flow is solved with periodic boundary conditions and a domain spanning $[0, 2\pi]$ in each direction, and the initial condition is

$$u = V_0 \sin(x)\cos(x)\cos(z), \tag{4.1}$$

$$v = -V_0 \cos(x)\sin(y)\cos(z), \tag{4.2}$$

$$w = 0, \tag{4.3}$$

$$p = \frac{\rho V_0^2}{16}(\cos(2x) + \cos(2y))(\cos(2z) + 2). \tag{4.4}$$

$$\tag{4.5}$$

The parameters $\rho$, $\mu$ and $V_0$ are chosen to give a Reynolds number $Re = \rho V_0 L/\mu = 1600$, where $L = 1\text{m}$. Plots of the initial condition (left) and close to the moment of highest turbulent intensity (right) are shown in Figure 4.3. The right plot is at $t = 9t_c$, where $t_c = L/V_0$ is the convective time scale.

The strategies used in the paper proved useful also to others. This prompted the author to write a short note, titled "Domain decomposition without the agonising pain" [243]. The code accompanying this note was submitted for review by the PETSc team (as a git pull request) and is now incorporated in PETSc as DMDA Tutorial 13 [244].

[243] Ervik, Å. (2015)

[244] Ervik, Å. (2015)

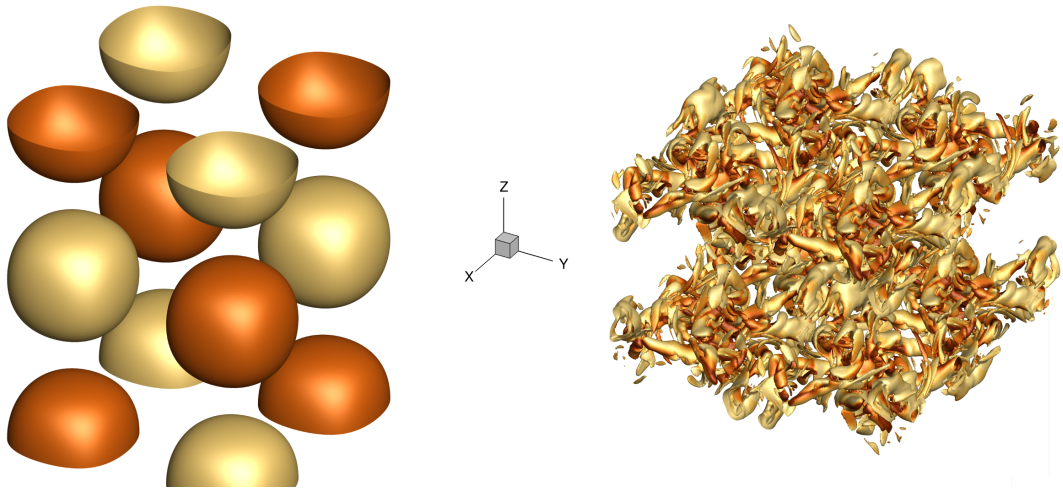



FIGURE 4.3: On the left, the initial condition, shown as two isosurfaces of the $z$-vorticity. On the right, $x$-vorticity isosurfaces coloured by vorticity magnitude. Note periodic boundary conditions.

One aspect of parallelisation which several people at the CFD 2014 conference expressed surprise about is the topic of memory bandwidth limitations. Memory bandwidth is the speed at which data in RAM can be loaded into the CPU. While the evolution in memory bandwidth over the years has been extensively documented by McCalpin[245], and the fact that memory bandwidth (rather than CPU core count) is the bottleneck for many algorithms has been popularised by the "seven dwarves" of Colella[246], it appears that many CFD practitioners are unaware of these facts.

CPU manufacturers are arguably part of the reason for this, since increasing core counts are heavily advertised, while memory bandwidth developments are relegated to tables in obscure specification documents.

People who are aware of the memory bandwidth problem talk about "scaling around" it. This is illustrated in Figure 4.4, where two benchmarks from the SINTEF cluster vsl176 are shown. On the left side of Figure 4.4, the memory bandwidth (tested with McCalpin's STREAM Triad benchmark) on a single dual-socket node is shown as a function of the CPU cores used. It is seen that for a memory-bandwidth limited simulation, there is no point in using more than ∼2 cores per node. The memory bandwidth on one node can be saturated by using two CPU cores. But $N$ nodes will have $N$ times the bandwidth of one node. On the right side of Figure 4.4, the same benchmark is shown for an increasing number of nodes using 4 cores per node; note that this is a log-log plot. In this figure the scaling is very linear with a slope of 0.88, illustrating that running a memory-bandwidth limited simulation on several nodes will scale around the limitation. This is provided the (InfiniBand) interconnect between nodes is as fast as the memory bandwidth, which cluster designers usually make sure is the case.

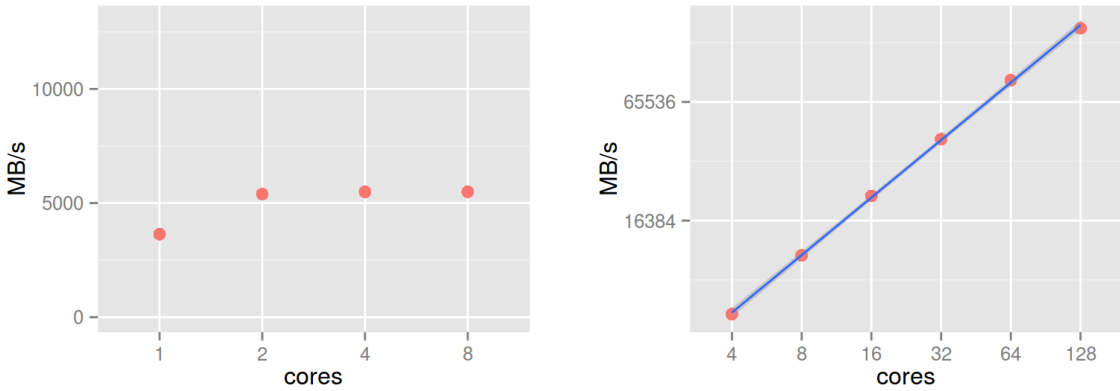



FIGURE 4.4: Comparison of memory bandwidth scaling on a single compute node, left, with the scaling across many nodes using 4 cores per node, right.

CHAPTER 5

# Summaries of thesis papers

Contents


5.1 Paper 1 . . . . . 71
5.2 Paper 2 . . . . . 75
5.3 Paper 3 . . . . . 77
5.4 Paper 4 . . . . . 79
5.5 Paper 5 . . . . . 83
5.6 Paper 6 . . . . . 87
5.7 Paper 7 . . . . . 91
5.8 Paper 8 . . . . . 93


> The best way to have a good idea is to have a lot of ideas.
>
> Linus Pauling

## §5.1 Paper 1

Ervik, Å., Lervåg, K. Y. & Munkejord, S. T. A robust method for calculating interface curvature and normal vectors using an extracted local level set. *Journal of Computational Physics*, **257**, 259–277. doi:10.1016/j.jcp.2013.09.053 (2014)

This paper presents the local level-set extraction (or LOLEX) method for the robust calculation of interfacial curvature when using the level-set method. As discussed in Chapter 3.2 and highlighted by the quote from Smereka[172], curvature can easily be calculated from the level-set function, except during a situation where the interface topology changes. This happens e.g. when two drops coalesce, or when a drop coalesces with a flat interface. Using the standard method, the curvature develops large unphysical oscillations in the moments before coalescence, as it is not possible for $\varphi$ to remain a signed distance function in the region between the drops. Since these curvature oscillations cause oscillations in the pressure difference across the interface, these numerical errors will lead to incorrect results. In

the most serious cases, the errors lead to an unphysical prevention of coalescence[247].

The LOLEX method is designed to solve this problem. It builds on the previous method by Salac *et al.*[248], and extends this method to considering only a local region of the interface. This locality has several important benefits: it means the method handles the entire class of cases where curvature calculation errors occur, as opposed to the method presented in Salac *et al.*[248] which assumes the curvature errors occur in the region between two separate bodies of fluid. The locality also means the method has low computational cost, since one can easily calculate the areas where the standard curvature calculation will give errors from the deviation of $\varphi$ from a signed distance function. Furthermore, the locality means the method is compatible with common parallelisation strategies. Finally, since the LOLEX method does not make use of curve fitting, in contrast to some methods developed to handle this problem[249] [237], the LOLEX method is easily extended to three-dimensional simulations, as demonstrated with a proof-of-concept in the paper.

In Figure 5.1 the advantage of the method over the standard curvature calculation is demonstrated. In this case, a water drop falls through a polybutene-decane mixture, and is about to merge with a pool of water. The colour in this figure indicates the pressure field. It is seen that with the standard method, significant pressure oscillations are present, while these are eliminated when using the LOLEX method.

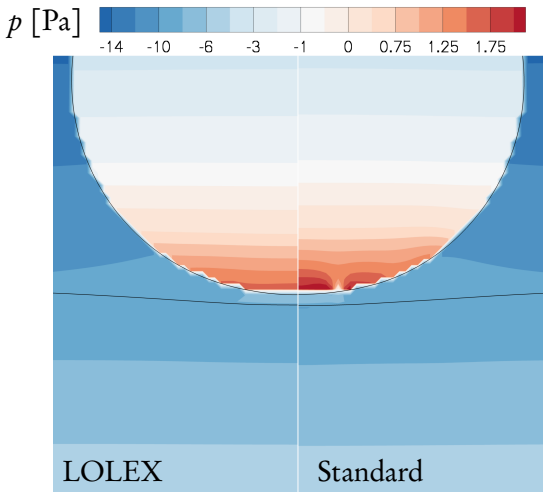



FIGURE 5.1: Water droplet falling through a mixture of polybutene and decane, about to merge with a water pool. The colours show the pressure field using the LOLEX method and the standard method at $t = 0.007$ s. (Reproduced from Paper 1.)

[247] Lervåg, K. Y. *et al.* (2013) [249] Macklin, P. *et al.* (2006) [237] Lervåg, K. Y. *et al.* (2013)

**Contributions to Paper 1**
The manuscript was written by me. The new method was developed and implemented into our in-house finite difference code by me, and most of the numerical results are due to me. Lervåg contributed with discussions during the development of the new method, designed the test case in Section 4.1, ran simulations for Section 5.2, created some of the result figures, and gave feedback on the manuscript. Lervåg also assisted in some of the programming efforts for initialising the test cases in Section 5. Munkejord contributed with discussions of the manuscript and some code testing.

Most of the work with the LOLEX method was done before starting the PhD project; the paper is an extension of my Master's thesis[250]. After completion of the Master's thesis, the method was tested on additional cases, some refinements of the method were undertaken, and the paper was written and submitted.

[250] Ervik, Å. (2012)

## §5.2 Paper 2

Ervik, Å., Munkejord, S. T. & Müller, B. *Extending a serial 3D two-phase CFD code to parallel execution over MPI by using the PETSc library for domain decomposition* in *Proceedings of the 10th International Conference on Computational Fluid Dynamics In the Oil & Gas, Metallurgical and Process Industries* (Trondheim, Norway, June 2014). arXiv: 1405.3805

In this paper we discuss the initial results with the code that I parallelised, which is an MPI-parallel 3D incompressible Navier-Stokes solver with single-phase and two-phase flow capabilities. This paper lays the foundations for extending the two-phase flow simulations in the present work, which are mainly performed in axisymmetry, to full three-dimensional configurations. This will allow the study of much more general cases, including the interactions of multiple droplets, or the effect of high-shear and turbulent flows on the deformations of a drop. In the paper, both weak and strong scaling are discussed. In a strong scaling study one measures the speedup when using more processes to solve a given problem, while in a weak scaling study one measures the time it takes to complete a simulation when both the number of processes and the amount of work are increased proportionally.

The strategy taken to parallelise the code was domain decomposition using MPI. This strategy is well-proven in the literature, and is expected to give good scaling results. As the name suggests, domain decomposition means that the computational domain is divided into smaller domains, which are then distributed to a set of processes each running the computations on its part. Several processes can run simultaneously on one multi-core CPU, but in general the processes will be distributed across many CPUs, to gain a larger memory bandwidth, as discussed in Chapter 4.3. This is illustrated in Figure 5.2, where a computational domain is divided into eight blocks. The colour indicates the pressure field, and the streamlines indicate the flow. The case here is a manufactured solution, so it has no physical meaning.

As discussed in Chapter 3.1, strong scaling will be limited (regardless of the parallelisation strategy) due to the elliptic nature of the pressure Poisson equation. The reported strong scaling in Paper 2 is in accord with this limitation: using 32 processes gives a speedup of 13 times faster than using just one process, which is good, but significantly below the ideal speedup of 32. The weak scaling is, however, much better. Again this is as expected from the literature. The weak scaling is measured by running a case on one process with $N = N_x \times N_y \times N_z$ total grid points, and then

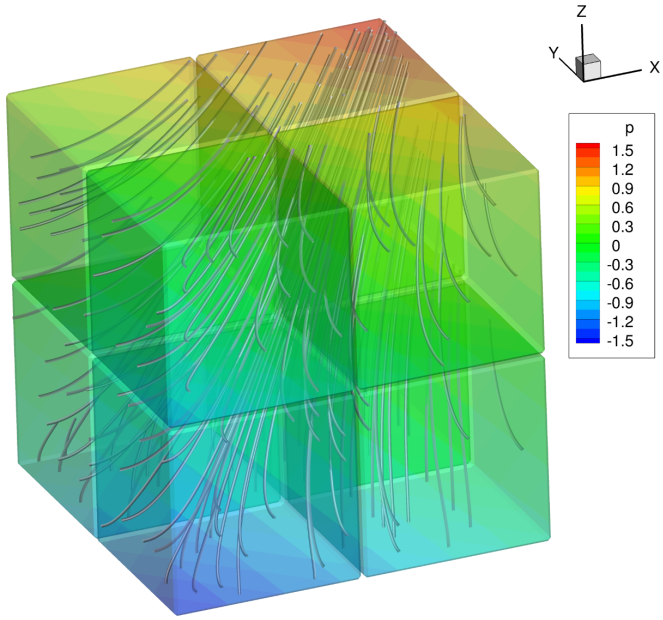



FIGURE 5.2: The computed solution after 0.031 s (100 time steps) on a $128^3$ grid run on 8 processors. The blocks show the decomposition of the domain, the pressure field is shown superimposed on these blocks, and the streamlines illustrate the flow. (From Paper 2.)

running a case with $2N$ total grid points on two processes, etc. and comparing the total runtime. It is found that while the total runtime increases when going from one to two processes, after this the total runtime stays fairly constant when increasing to four and then to eight processes.

**Contributions to Paper 2**

The manuscript was written by me. I wrote the code, which is based on a previous 3D serial code written by Munkejord and others. I performed the simulations and analysis reported in the manuscript. Munkejord contributed with discussions around the details of the staggered grid implementation, and with general discussions of the manuscript and the reported results. Müller contributed with discussions of the manuscript and the reported results.

## §5.3 Paper 3

Ervik, Å., Hellesø, S. M., Munkejord, S. T. & Müller, B. *Experimental and computational studies of water drops falling through model oil with surfactant and subjected to an electric field* in *Proceedings of the IEEE 18th International Conference on Dielectric Liquids* (Bled, Slovenia, July 2014). doi:10.1109/ICDL.2014.6893172

In this paper we discuss the results of experiments and axisymmetric two-phase flow simulations of five cases of electrohydrodynamic two-phase flow, and direct comparisons of these results. We consider a single salt-water drop immersed in a pure alkane-based oil with added surfactants and subjected to a strong electric field. To be specific, the drop phase contained Milli-Q purified water with 3.5 % (by weight) NaCl, and the oil phase consisted of Marcol 52 (ExxonMobil) to which Span 80 surfactant (Sigma Aldrich) was added at concentrations from 0.001 to 0.02 % (by weight). This particular surfactant is chosen because it is non-ionic (thus does not increase the oil-phase conductivity) and because it has emulsion stabilising properties similar to those of crude oil components[252]. The interfacial tension as a function of concentration was measured with a DuNüoy ring tensiometer, and a Langmuir equation of state was fitted to these measurements. This equation of state was then used in the simulations.

In the experiments, drops were produced with a syringe, and allowed to fall between two horizontal electrodes, to which a potential difference was applied in the form of square pulses. See Figure 1 in the paper for a schematic. The applied voltages produce a high strength electric field of the order of 500 V/mm, and the drop diameters are between 0.5 and 1 mm. A high-speed near-infrared camera was used to record videos of the drop deformations, and from these the drop deformations were extracted as the quantitiy $a/b$, where $a$ is the length of the vertical droplet axis and $b$ the horizontal droplet axis (see Figure 2 in the paper).

The exact same cases were then simulated using the axisymmetric two-phase flow code discussed in Chapter 4.3, and $a/b$ was computed also from the simulations. An example of the comparison in shown in Figure 5.3. In this figure, the simulation results are shown both with and without surfactants, and the theoretical expression by Taylor[6] for the equilibrium deformation is also shown at the right-hand side. It is seen that the simulation with surfactants is in good agreement with the experimental result, while the simulation without surfactant predicts a larger deformation, and is in agreement with the result by Taylor.

[252] Santini, E. *et al.* (2007)

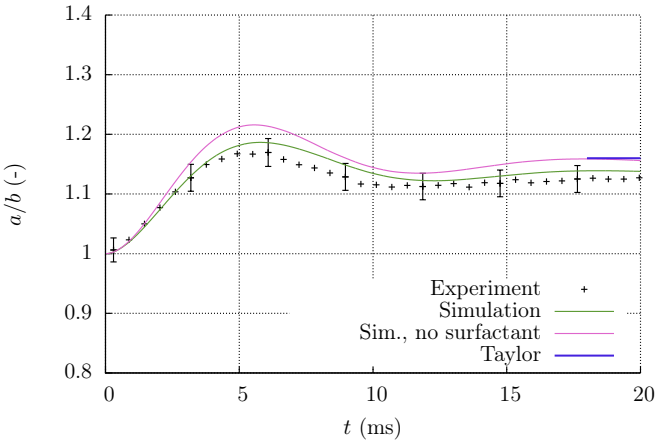



FIGURE 5.3: **Case 1:** Simulation and experimental results for the 958 $\mu$m diameter water drop falling through Marcol with 0.015 wt% Span 80 subjected to a 400.0 V/mm field. (From Paper 3.)

**Contributions to Paper 3**

The manuscript was written by me. I performed the numerical simulations and created the plots, except for the experimental schematic (Fig. 1) and the plot of drag coefficient vs. Reynolds number (Fig. 3) which were created by Hellesø. Hellesø performed the experiments and the post-processing of the high-speed camera footage, and contributed to the description of the experimental technique. Hellesø, Munkejord and I discussed the selection of experimental parameters and fluid combination. Munkejord performed the fit of the Frumkin equation to the experimental data, and contributed with testing and improvement of the code. Munkejord and Müller contributed with discussions of the manuscript and the reported results.

## §5.4 Paper 4

Ervik, Å., Penne, T. E., Hellesø, S. M., Munkejord, S. T. & Müller, B. Influence of surfactants on the electrohydrodynamic stretching of water drops in oil. *Submitted to Physical Review Fluids*, (2016)

This paper is a significant extension of Paper 3 (where five representative cases were studied). In Paper 4 we perform a parameter study covering a much wider range, using 44 cases in the simulations and almost 300 cases in the experiments. We study the effects of drop radius, surfactant concentration and electric field strength on the drop deformations. The fluids, experimental and numerical techniques are the same as in Paper 3. The fitting of the Langmuir equation of state to measurements of interfacial tension versus surfactant concentration was redone, and an improved fit was obtained.

In this paper, increasing  and decreasing  voltage pulses were used in the experiments. This is the only way in experiments to study the effect of the electric field strength at constant drop diameter, since no two drops produced will have the exact same diameter. Also, a moving stage setup was used to maintain the drops in the camera field-of-view while they fell through the oil, allowing for longer voltage pulses to be studied. Up to ten deformations of the same drop were possible with this setup. By using both increasing and decreasing pulses, it was possible to study the possible influence of previous deformations, i.e. a hysteresis effect as reported e.g. by Peltonen *et al.* [253] for flat fluid interfaces with essentially the same fluid combination. The observation from experiments was that at high surfactant concentrations, drop deformations increasing in magnitude from small to large caused a significant hysteresis effect, while with deformations decreasing in magnitude from large to small, no hysteresis was observed. This goes against the hypothesis put forward by Peltonen *et al.* [253], who proposed that earlier deformations cause some surfactants to detach into the water phase. If this were the case, hysteresis should be stronger for the decreasing deformations. The observations here find the opposite to be true, which suggests that alternative explanations, in particular rearrangement of the surfactants at the interface by the initial small deformations, are a more likely explanation.

Using the larger data set obtained in this study, an analysis was performed of the influence of the surfactants on the damping of the initial oscillations after the electric field is applied. This damping was characterised by the parameter

$$\Omega = \frac{(a-b)_{\text{peak}} - (a-b)_{\text{static}}}{(a-b)_{\text{static}}}, \tag{5.1}$$

where $(a-b)_{\text{static}}$ refers to the static deformations observed e.g. in Figure 5.3 after 20 ms, while $(a-b)_{\text{peak}}$ refers to the peak deformation observed after 5 ms. It is important to note, however, that here the difference between the two axes, $(a-b)$, is used (as opposed to the ratio $(a/b)$ discussed previously). This is because $(a-b)$ is directly proportional to the amplitude of the second spherical harmonic, assuming this is the dominant mode of oscillation. Accordingly, if one assumes the classic results by e.g. Lamb[87] for the damping of such oscillations to hold in this case, $(a-b)$ can be directly related to the damping ratio of a harmonic oscillator, $\lambda$, as

$$\lambda = \sqrt{\frac{\ln(\Omega)^2}{\pi^2 + \ln(\Omega)^2}}, \tag{5.2}$$

using here standard results from control theory; see e.g. the derivation on p. 172 of the book by Ogata[254].

This damping ratio $\lambda$ was computed from both the simulation and experimental results. We plot $\lambda$ versus the dimensionless electric field strength given by

$$\zeta = \bar{E}\sqrt{\epsilon\epsilon_0 D/\gamma} = \sqrt{Ca_E}, \tag{5.3}$$

i.e. the square root of the electric capillary number, which was introduced by Taylor to describe the drop deformations. In this expression $\bar{E}$ is the electric field strength far away from the drop, $D$ is the drop diameter, $\gamma$ is the equilibrium interfacial tension, and $\epsilon\epsilon_0$ is the electric permittivity of the oil. The plot for the simulation results is shown in Figure 5.4, the experimental plot is similar. In this figure, the surfactant concentration, drop radius and electric field strength are indicated by point shape, point size, and point colour, respectively. Cases with identical drop size and electric field strength, but varying surfactant concentration, are connected with lines. It is seen that the effect of surfactants is two-fold: addition of small concentration of surfactants increases the damping $\lambda$ significantly, but has little effect on the equilibrium interfacial tension on the drop (which enters into the expression for $\zeta$). But when increasing the surfactant concentration to a value of 0.016 wt%, closer to the critical micelle concentration of 0.020 wt%, the damping does not increase much further, while the interfacial tension is significantly decreased, causing an increase in $\zeta$ and in the drop deformation.

To further support this hypothesis, detailed snapshots from the simulations were considered for four cases with identical drop size

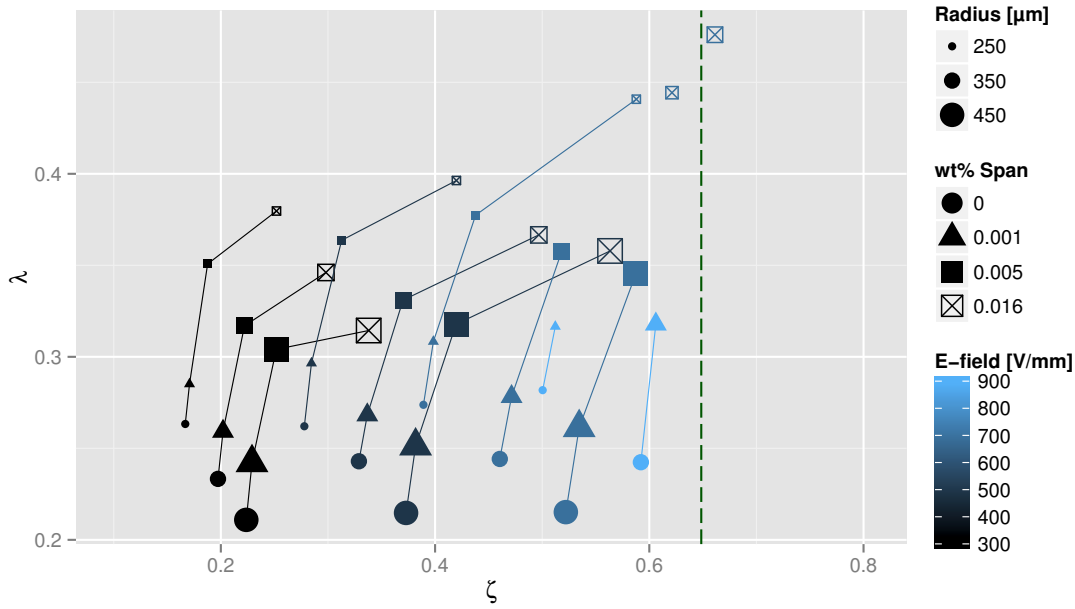



FIGURE 5.4: The damping ratio of oscillations, $\lambda$, versus dimensionless electric field strength, $\zeta$. Dashed line: static deformation limit. (From Paper 4.)

$D = 500\ \mu$m and electric field strength $\bar{E} = 700$ V/mm, but increasing surfactant concentration. The values of $\zeta$ are 0.39, 0.40, 0.44 and 0.59 in order of increasing surfactant concentration. These snapshots are shown as the four quadrants in Figure 5.5. In this plot, the increase in Marangoni force when going from zero to low surfactant concentration, illustrated by the tangential component of the black vectors at the interface, is clearly seen. One can also see the decrease in interfacial tension at the highest surfactant concentration (lower left quadrant), and that this allows for a stronger flow field and thus larger deformation. These snapshots are all taken at $t = 1$ ms, which corresponds to half-way to the first peak of deformation.

### Contributions to Paper 4

The manuscript was written mainly by me. I performed the numerical simulations and analysis of numerical and experimental results, and created the plots, again except for the experimental schematic which was created by Hellesø. Penne performed the experiments and the post-processing of the high-speed camera footage. Penne also contributed to the description of the experimental technique in the manuscript. Hellesø, Munkejord and Müller contributed with discussions of the manuscript and the reported results.

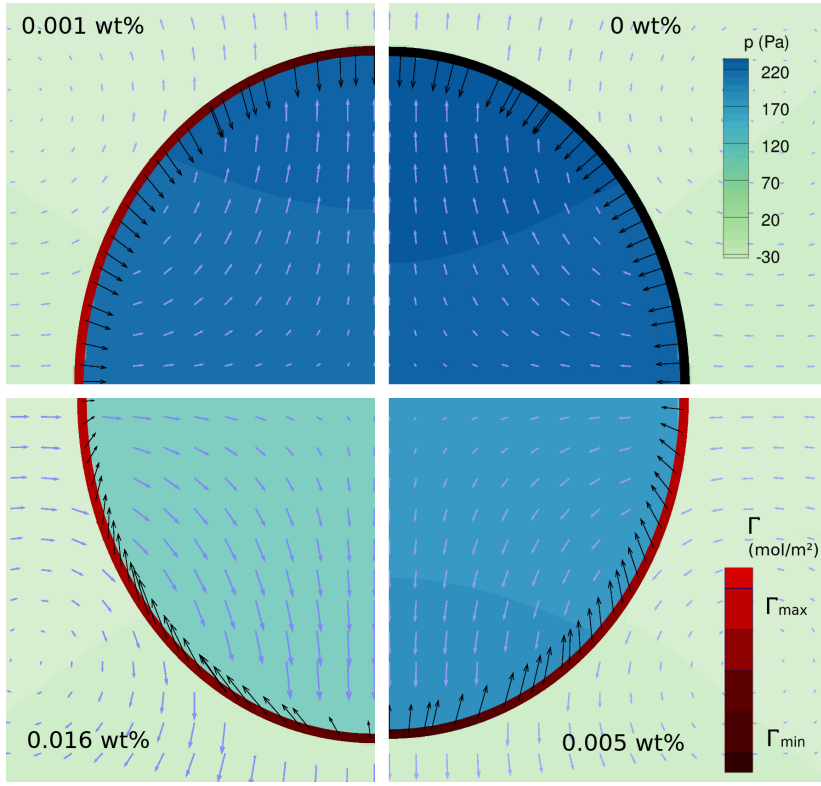



FIGURE 5.5: The effect of surfactant concentration on normal and tangential interfacial stress. Comparison of the pressure field (blue/green), the vector $\gamma\kappa\mathbf{n} - 100\nabla_s\gamma$ (black vectors), the flow field (sky-blue vectors) and the surfactant concentration at the interface (black for zero concentration, redness indicating concentration). The quadrants show the four different bulk concentrations considered. (From Paper 4.)

## §5.5 Paper 5

Ervik, Å. & Bjørklund, E. The admissible surfactant distributions and velocities for small falling drops. *Submitted to Journal of Fluid Mechanics*, (2016)

This paper discusses the effects of surfactants on a single falling drop at low Reynolds- and Weber-numbers from the theoretical perspective. The purpose of this work is to present a new approach which explains the transition from drops falling like hard spheres, to drops falling according to the Hadamard-Rybzynski solution, as the drop size is increased. Extending the derivation by Chang *et al.*[256] to include also the normal interfacial stresses as boundary conditions for the Stokes equation at the drop interface, we are able to circumvent the need to solve a transport equation for the surfactant in order to obtain a solution for the interfacial tension profile and the flow field. As pointed out e.g. by Leal[82], this has been a major hurdle in previous work which led to the stagnant-cap model:

> It is not generally possible to obtain analytic solutions of the resulting problem because of the complexity of the surfactant transport phenomenon and the coupling between surfactant transport and fluid motion. (L. Gary Leal)

Another, perhaps even more important difficulty with using the surfactant transport equation as a starting point is that the expression includes the tangential velocity along the interface. However, this velocity must be zero for drops that fall like hard spheres. This leads to paradoxes and inconsistencies in the typical approach in the literature. Consider e.g. Equation 7-270 on page 497 in the book by Leal[82], which is derived assuming the surfactant is insoluble and that the interfacial Péclet number is $Pe_s = 2R|\mathbf{u}_s|/D_s \gg 1$, where $R$ is the drop radius, $\mathbf{u}_s$ is the velocity field on the interface, and $D_s$ is the surface diffusivity of the surfactant. The equation in Leal's book reads

$$\boldsymbol{\nabla}_s \cdot (\mathbf{u}_s \Gamma) \approx 0 \tag{5.4}$$

where $\Gamma$ is the surfactant concentration. From this the classic stagnant cap result is obtained, namely that a part of the interface has $\mathbf{u}_s = 0$ and another part has $\Gamma = 0$. Here comes the paradox: if $\mathbf{u}_s = 0$ in the region where surfactants are found, then $Pe_s = 0$! This clearly violates the initial assumption $Pe_s \gg 1$, so the result is not self-consistent.

In the work in Paper 5, these difficulties are circumvented by not considering any transport equation or surfactant concentration

[82] Leal, L. G. (2007)

at all, but merely applying the appropriate boundary conditions at the interface. Interestingly, this gives an analytical solution where the expression for the interfacial tension along the interface has the same form as that obtained e.g. by Levich[81], who solves a transport equation under some simplifications, e.g. the assumption that the variation in surfactant concentration along the drop is small. Unlike these classic derivations of the stagnant-cap model, our approach makes fewer assumptions and is self-consistent.

We further consider the problems with assuming a surfactant transport equation, by tracing this equation back to its roots, the Langevin equation. Considering the link between this equation and the transport equation, we show that results obtained assuming a surfactant transport equation can give either a non-zero force from the interface (acting to slow the fall of the drop), or a zero tangential velocity at the interface (as when the drop falls like a hard sphere), *but not both of these simultaneously*. This leads to the observation that the assumption of a surfactant transport equation cannot generally lead us to any true conclusions (to paraphrase the previous quotation by the reverend Samuel Vince).

To proceed from the Langevin equation, we discard the notion that the force on the surfactant molecules can be determined from a Stokesian friction, and instead consider the surfactants to interact as if they were an ideal gas. Such an assumption may seem odd, but it is in fact a classical approach when it comes to surfactants, going back to the work which earned Langmuir[257] the 1932 Nobel Prize in Chemistry. Assuming ideal gas behaviour, we derive an expression which allows for both a force from the interface acting to slow down the drop, and an interfacial velocity which is simultaneously zero. This leads to the result that the critical radius $R_c$, below which drops fall like hard spheres, is directly proportional to the *interfacial* surfactant concentration.

Using this alternative approach, which we dub the continuous-interface model, we consider the experimental results obtained by Griffith[258] for the transition between hard-sphere and fully-fluid interfacial behaviour at different surfactant concentrations. Griffith considers ethylene glycol drops falling in mineral oil to which Aerosol C61 surfactant is added. By non-linear curve fitting, the critical radius $R_c$ is determined for each bulk concentration, as shown on the left of Figure 5.6.

As mentioned, the continuous-interface model predicts that the critical radius is directly proportional to the interfacial surfactant concentration. To test this prediction, we plot the critical radii versus

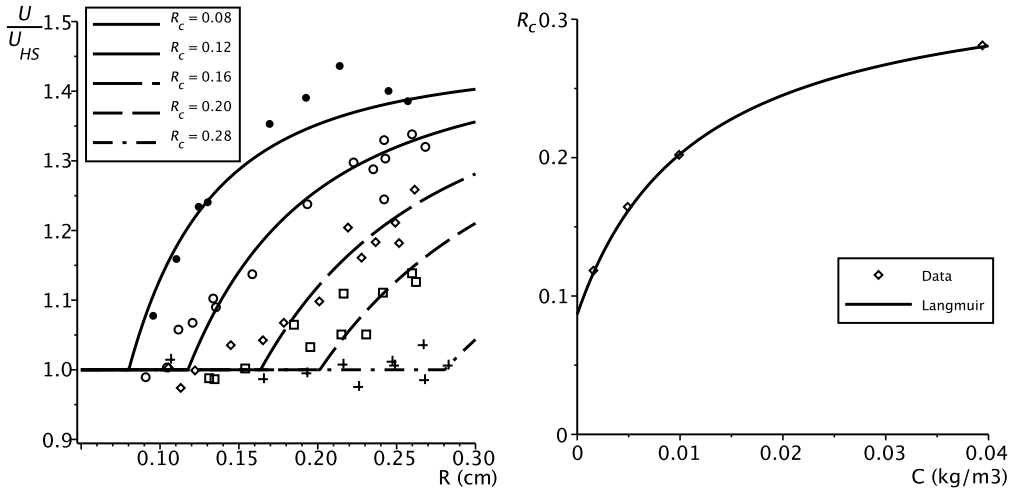



FIGURE 5.6: Left: nonlinear curvefit to the raw data by Griffith[258]. Corresponding values for $R_c$ shown in the caption. Right: $R_c$, obtained from the left plot, plotted against bulk concentration of surfactants. The line is a Langmuir isotherm fitted to these points, with excellent agreement. (From Paper 5.)

the bulk surfactant concentration, and demonstrate that these points collapse perfectly onto a Langmuir equation of state, shown on the right of Figure 5.6. Since such an equation of state gives the relation between the interfacial and the bulk surfactant concentration, this confirms the prediction by the continuous-interface model. Note that the equation of state here does not go to zero at zero bulk concentration, since the results by Griffith indicate that the oil contains some surfactants even when none are explicitly added. This is as expected, cf. the discussion in Chapter 2.3.

Finally, it is noted that the relation between critical radius and interfacial surfactant concentration may have applications to the direct experimental measurement of this quantity, which otherwise requires complicated and expensive techniques such as small-angle neutron scattering[259].

**Contributions to Paper 5**

The manuscript was written mainly by me. I performed the literature study. Bjørklund performed the theoretical analysis and created the plots of the analytical solutions. I contributed with discussion of the theoretical analysis, including the underlying assumptions and the implications of the results.

[259] Verruto, V. J. *et al.* (2008)

## §5.6 Paper 6

Ervik, Å., Serratos, M. G. J. & Müller, E. A. raaSAFT: a framework enabling coarse-grained molecular dynamics simulations based on the SAFT-$\gamma$ Mie force field. *Submitted to Computer Physics Communications*, (2016)

In this "Computer Programs in Physics" paper we present raaSAFT, a framework for running molecular dynamics simulations using the SAFT-$\gamma$ Mie approach. The paper briefly introduces the theory behind the approach, with references to earlier publications[201] [202] [203] [204] [205] [206] [200] [207] [208]. We then review the physical unit convention used in raaSAFT, describe how the software is installed, and give a detailed walkthrough of the user interface for running a simulation. Thorough descriptions are also given of how models for chemical compounds are implemented, first in the simple homonuclear case, and subsequently for the more complicated case of a heteronuclear model, such as for a surfactant where the head and tail are constructed from different beads. We give a brief description of how raaSAFT interacts with Bottled SAFT (Paper 7).

Finally, we present several case studies of simulations performed using raaSAFT. The simplest cases include liquid-liquid equilibrium between octane and water, and the simulation of liquid toluene for the weak scaling study. To fully demonstrate how raaSAFT enables large simulations of highly complex systems with few lines of code, we consider the phase behaviour of polystyrene in heptane at high pressure (30 bar) and varying temperature (310, 420 and 530 K). Using simulations with 300 polystyrene molecules and 42 700 molecules of the explicit solvent heptane, corresponding to 1.2 million atoms, we reproduce the experimentally observed[261] temperature-dependent solubility of polystyrene in this system. To reach equilibrium, the simulation at the lowest temperature has to run for 400 nanoseconds (or 40 million timesteps). We argue that if this system were to be studied using atomistic simulations, even with parallelisation and using multiple GPUs, it would take more than a year from simulation start to finish. With raaSAFT, running on four Nvidia Tesla K20 GPUs, the runtime is 135 hours. A snapshot of this system is shown in Figure 5.7, for 310 K where the polystyrene is insoluble (top) and for 420 K where the polystyrene is soluble (bottom).

[201] Avendaño, C. *et al.* (2011)
[202] Avendaño, C. *et al.* (2013)
[203] Lafitte, T. *et al.* (2012)
[204] Müller, E. A. *et al.* (2014)
[205] Herdes, C. *et al.* (2015)
[206] Herdes, C. *et al.* (2015)
[200] Lobanova, O. *et al.* (2015)
[207] Lobanova, O. *et al.* (2016)
[208] Theodorakis, P. E. *et al.* (2015)
[261] Cowie, J. *et al.* (1983)

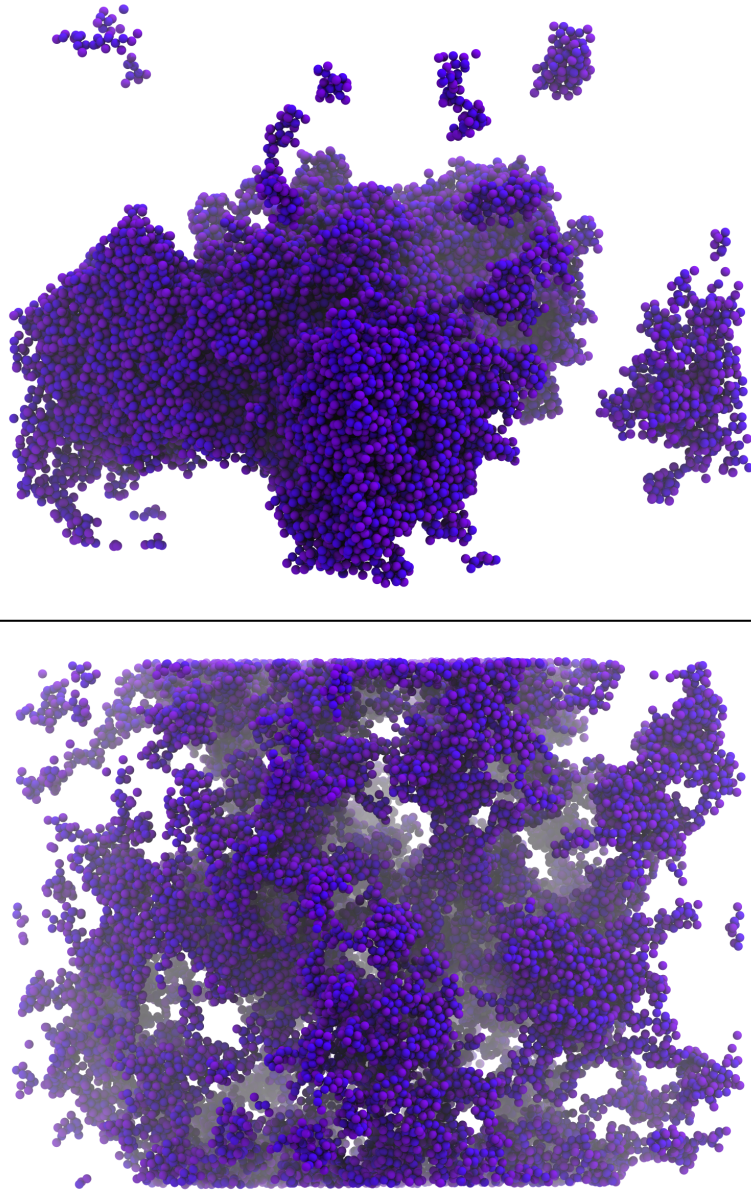


FIGURE 5.7: Snapshots of equilibrated configurations from a simulation of polystyrene in heptane at 310 K (top) and 420 K (bottom). The blue and purple beads represent the polystyrene molecules. Heptane molecules are not shown for clarity. It is seen clearly that the polystyrene is dissolved at 420 K while it is clustered at 310 K. At the top of the figure, individual polymers can be seen. The individual beads seen close to the edges are due to the periodic boundary conditions. (From Paper 6.)

**Contributions to Paper 6**
I wrote the manuscript and the code and performed the simulations. Jiménez-Serratos developed the polystyrene model and the code to analyse the simulations using this model, and contributed with discussions of the paper. Müller contributed with discussions of the manuscript and the reported results.

## §5.7 Paper 7

Ervik, Å., Mejía, A. & Müller, E. A. Bottled SAFT: a web app providing SAFT-$\gamma$ Mie force field parameters for thousands of molecular fluids. *Submitted to Journal of Chemical Information and Modelling*, (2016)

In this paper we present Bottled SAFT, a web application which provides force field parameters for more than 6000 molecular fluids, from the simple linear alkanes, to molecules important in the pharmaceutical industry such as aspirin and ibuprofen. The database is searchable by name, chemical formula, or CAS number.

In the paper, we give again a brief review of the SAFT-$\gamma$ Mie approach, as well as an overview of the tools used to implement Bottled SAFT. We proceed to present the details of the implementation, including how the data from Yaws[223] was run through the correlation and results outside the scope of the correlation were discarded. We also present the components which make up the web application, including code which shows how the Flask framework enables the implementation in a simple and understandable fashion.

Using the Locust framework the performance of Bottled SAFT is tested. It is found that the application can sustain a load of 70 000 database lookups per hour, as well as 800 000 views of the front page per hour, which should be more than sufficient for a web application serving the molecular modelling community.

To demonstrate the power of Bottled SAFT as a tool, we consider the case of ternary liquid-liquid equilibrium between octane, benzene and sulfolane. This system is very important in the chemical processing industry, where sulfolane is used to separate aromatic compounds, such as benzene and toluene, from the aliphatic-aromatic mixture that comes from petroleum distillation. Using the force field parameters provided by Bottled SAFT, we first tune the cross-interaction between octane and sulfolane to obtain agreement with the experimentally measured mutual solubilities of these two liquids at a temperature of 373.15 K. We then study the full ternary system at a temperature of 403.15 K, for which experimental measurements are available due to Lee *et al.* [263]. Starting at a molar composition of 20/40/40 benzene/octane/sulfolane, we obtain a liquid-liquid split where the molar fractions in both phases match the experimental measurements extremely well. This is shown in Figure 5.8.

**Contributions to Paper 7**

I wrote the web application and the manuscript and ran the simulations. Mejia contributed with testing of the application and

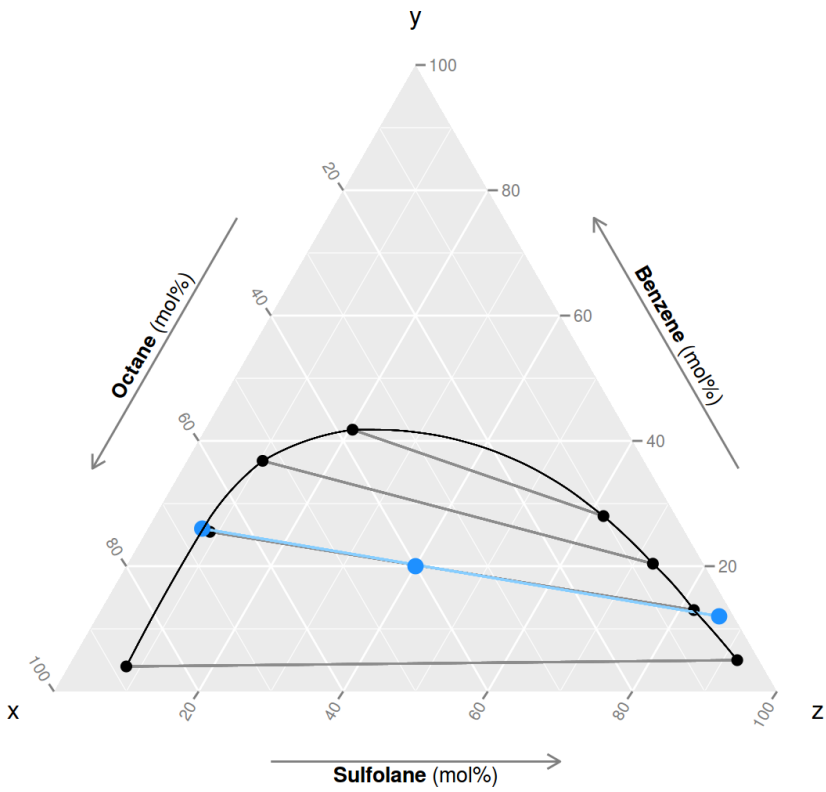



FIGURE 5.8: The ternary diagram for the octane-benzene-sulfolane system. Black points and grey lines: experimental results by Lee *et al.* [263]. Blue points and blue line: simulation results. (From Paper 7.)

provided the codes for the M&M correlation. Müller conceived of the original idea and guided the implementation.

## §5.8 Paper 8

Ervik, Å., Lysgaard, M. O., Herdes, C., Jiménez-Serratos, G., Müller, E. A., Munkejord, S. & Müller, B. A multiscale method for simulating fluid interfaces contaminated by large molecules such as asphaltenes. *Submitted to Journal of Computational Physics*, (2016)

In this paper the multiscale method developed here for the simulation of asphaltene-covered interfaces is presented. Preliminary results from this work were selected for an oral presentation at the 2015 PetroPhase conference[264].

We review the properties and modelling of fluid interfaces, and highlight the significant departures from surfactant-covered interfaces which occur when one considers asphaltene-covered interfaces. We then proceed to present the theory and numerical methods used as parts of the multiscale approach, in particular the hybrid level-set/ghost-fluid/immersed-boundary method. The multiscale approach is then summarised, using Figure 5.9.

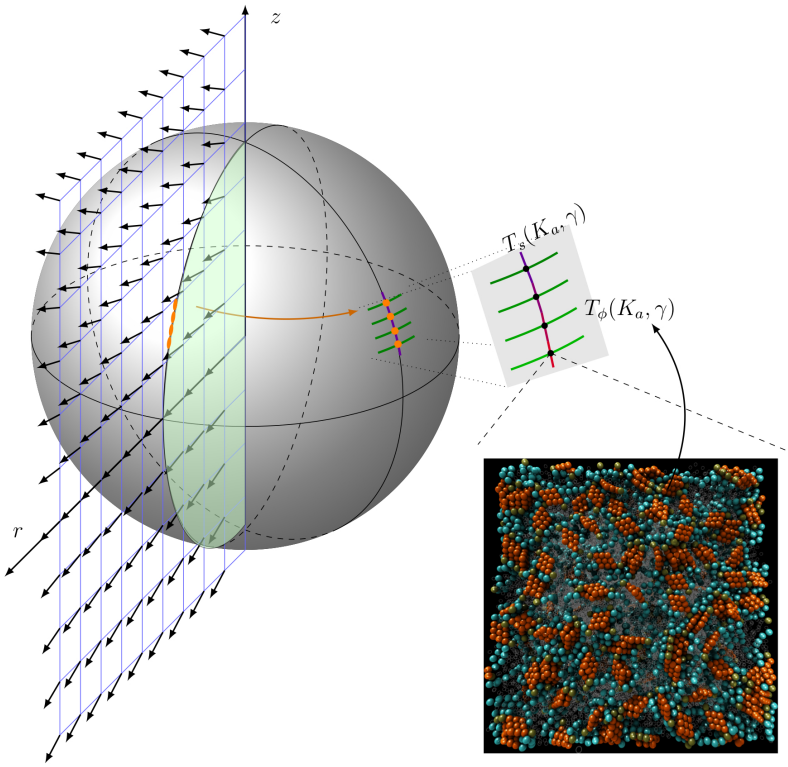



FIGURE 5.9: Illustration of the multiscale method. On the Eulerian $(r,z)$ grid, the flow field $\mathbf{u}$ (vectors) and the level-set function $\varphi$ representing the green-shaded portion of the droplet are shown. Some of the Lagrangian immersed-boundary points are shown in orange, with lines indicating the tensions $T_s$ and $T_\phi$. For a tiny patch on the interface, we compute the interfacial properties $K_a$ and $\gamma$ using molecular dynamics simulations (lower right corner). These properties are used in the calculations of the tensions. (From Paper 8.)

Using the SAFT-$\gamma$ Mie approach at the nanoscale, the effect of asphaltenes at the interface is studied for two different model

[264] Ervik, Å. *et al.* (2015)

asphaltenes. These differ in the force field and coarse-graining level for the beads which make up the aromatic core of the molecules. The second model asphaltene, dubbed the APCE model, is developed partly in this work. We demonstrate that this APCE model gives an interfacial behaviour in contact with water which is consistent with experimental results using sum frequency generation spectroscopy; i.e. that the aromatic cores align with the water interface, while the aliphatic tails point into the oil phase. To our knowledge, this is the first model asphaltene in the literature which shows this behaviour in molecular dynamics simulations. The interfacial tension $\gamma$ as well as the interfacial elasticity $K_a$ are computed from the molecular simulations.

Using the developed hybrid method at the continuum scale, two cases of drop deflation are simulated using representative values of the interfacial tension and elasticity. Drop deflation is interesting because of the "crumpled drop" phenomenon which occurs. This phenomenon is not observed in water-oil-surfactant systems, but occurs for asphaltene-covered interfaces. It is thus useful for qualitative comparisons with experiments. We consider two cases which have been studied in the experimental literature, namely the micropipette case and the pendant drop case. We point out that in the former situation the Eötvös number is negligible, meaning that buoyancy is negligible, while in the latter situation the Eötvös number is $Eo \approx 0.5$, meaning that the effect of tension in the interface and the effect of buoyancy are approximately in balance. Using axisymmetric simulations, we demonstrate that the characteristic crumpling around the neck, observed in pendant drop experiments, is due to an anisotropic total tension caused by the interfacial elasticity. The simulation and experimental results are compared in Figure 5.10. Using two-dimensional simulations, we demonstrate that the uniform crumpling seen in the micropipette case, where buoyancy is negligible, is caused by the tension being equal across the drop. The simulation and experimental results are compared in Figure 5.11.

**Contributions:**
I wrote most of the manuscript, performed the molecular dynamics simulations and analysis of these, and performed some of the continuum simulations. Lysgaard and myself jointly developed the hybrid continuum method. Lysgaard wrote the code for this method, performed most of the simulations with this method, and contributed with most of the description of this method in the manuscript. Herdes and Jiménez-Serratos and E. A. Müller

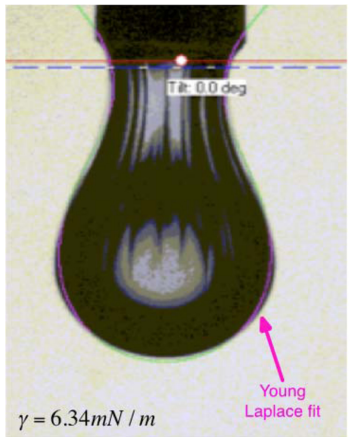
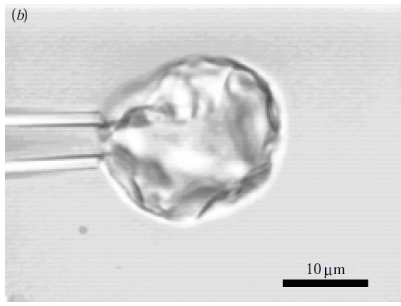



FIGURE 5.10: Right: raytrace of the drop profile from simulations, revolved around the symmetry axis and with crumples inserted into the region where $T_\phi = 0$. Left: experimental result showing a deflated asphaltene-stabilised water drop in oil. (Left figure reprinted with permission from: V. Pauchard, J. P. Rane, S. Banerjee, Asphaltene-laden interfaces form soft glassy layers in contraction experiments: A mechanism for coalescence blocking, Langmuir 30 (2014) 12795–12803. Copyright (2014) American Chemical Society.)



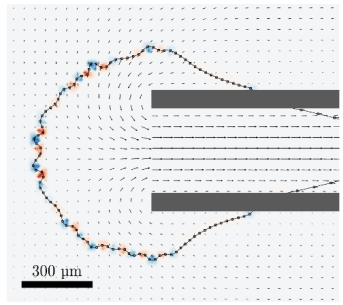




FIGURE 5.11: Right: two-dimensional simulation, with red and blue colours indicating interfacial curvature. Velocities are plotted for every 5th grid point and every 10th Lagrangian point is plotted. Left: experimental result showing the crumpled drop attached to a micropipette. (Left figure reprinted with permission from: A. Yeung, T. Dabros, J. Czarnecki, J. Masliyah, On the interfacial properties of micrometre-sized water droplets in crude oil, Proceedings of the Royal Society of London A (1999) 3709–3723.

contributed with the development of the SAFT force fields and molecular model for asphaltene molecules. Herdes, Jiménez-Serratos, Lysgaard, Munkejord and B. Müller contributed with discussions of the manuscript and the reported results.

CHAPTER 6

# Conclusions and future outlook

Contents


6.1 Conclusions . . . . . 97
6.2 Future outlook . . . . . 98


Three-dimensional hydrodynamics can eat up an arbitrary capacity on any computer we're ever likely to see. So you have to be clever.

Berni Alder

## §6.1 Conclusions

The separation of water from crude oil initially sounds like a simple task; as children we are taught that water and oil do not mix. While indeed these fluids are immiscible, they can still form emulsions, in which tiny water drops are scattered throughout the oil phase, and chemicals adsorbed at the interface prevent drop coalescence and thus stabilise the emulsion. As reviewed in Chapter 1, much effort has gone into improving techniques for the removal of water from crude oil during the last century, and yet there are still open questions.

Using the multiscale approach developed in this thesis, it is attempted to answer at least some questions regarding water drops in crude oil. In this approach, coarse-grained molecular dynamics simulations using the SAFT-$\gamma$ Mie force field are used to estimate the properties of oil-water interfaces covered with asphaltenes. In particular, the interfacial tension $\gamma$ and the elasticity $K_a$ are computed. These properties are subsequently used in two-phase flow simulations, using a hybrid level-set/ghost-fluid/immersed-boundary method developed in this work to handle the complex interface.

In particular the crumpled drop phenomenon is considered, and the physical mechanisms underlying this phenomenon are revealed,

both for tiny emulsion drops and for the larger pendant drops. The multiscale approach holds promise for use in future studies, since it offers the possibility of detailed hypothesis testing with regards to the effect of crude oil composition, the effect of asphaltene architecture and polydispersity, the effect of resins or of demulsifiers, and so forth.

Using existing numerical methods, together with experimental results, the effects of electric field on drops in the surfactant/oil/water model system is considered. Building on a large parameter study, the effect of surfactants on the transient damping of the electrohydrodynamic drop deformations is analysed in detail. At high surfactant concentration, hysteresis in the deformations is observed in the experiments, but only for deformations of increasing magnitude. It is suggested that rearrangement at the interface is responsible for this effect.

While the coalescence blocking properties of surfactants and other interfacially active molecules are essential for emulsion stability, the reduced terminal velocity of contaminated drops is also an important effect. Theoretical work presented in this thesis proposes a new model which explains the observed transition in terminal velocity as drops increase in size. This model is an improvement over the existing stagnant cap model in the literature, which has been the *status quo* for sixty years. These results may also have interesting implications for the measurement of interfacial surfactant concentrations.

Molecular simulations using the SAFT-$\gamma$ Mie approach form an essential part of the multiscale approach. Two computational tools for molecular dynamics simulations have been developed as part of this work, namely raaSAFT and Bottled SAFT. Together, these enable the study of a wide range of fluid systems, by providing accurate force field parameters for 6000+ fluids (Bottled SAFT) and an easy-to-use implementation of the SAFT-$\gamma$ Mie approach, which also leverages the computational power of GPUs (raaSAFT).

Taken together, the tools and methods developed and the results obtained in this work bring us closer to a detailed understanding of the physical phenomena which govern the process of water-oil emulsion separation, be it using electric fields or other means.

## §6.2 Future outlook

Working on this thesis I have learned many things, not the least of which is that there is still a very great deal to learn, investigate and discover. Here I provide some ideas, hints and speculations about

future research directions, in the hope that they may be useful to others.

**Suggestions for incompressible flow simulations**

There are many interesting avenues of investigation for incompressible flow. An interesting open question (as far as the author is aware) even for single phase flow is whether it is possible to construct a second order projection method using an explicit RK method. Another path to a higher-order method is by building on a very interesting recent paper by Lalanne *et al.* [265], which has paved the way for a fully implicit projection method also for two-phase flows using the ghost-fluid method. Previous approaches have been limited to using the less accurate continuous surface force method for the viscous term in order to obtain a fully implicit method.

Combining the approach by Lalanne with for instance the BDF2 time integration method and the level-set/ghost-fluid method would give an overall method with highly desirable properties: implicit time integration allowing large time steps (particularly important for low Reynolds number flows), second-order in time (at least for single-phase flow), excellent stability properties, sharp handling of the interface, and only one pressure Poisson equation to solve per time step.

As mentioned in Chapter 3.1, and demonstrated in Paper 2, strong scaling is an issue when solving the pressure Poisson equation. Much research has gone into fast solvers and preconditioners for this equation. A very interesting recent development is the work by Patil *et al.* [266], where the lattice Boltzmann method is employed together with a multigrid method to solve the two-dimensional Poisson equation with excellent scaling results. It is well known that the lattice Boltzmann method is very amenable to acceleration using GPUs. On the other hand, it is the author's understanding of the consensus in the literature that GPUs are unable to provide significant speedups for the incompressible Navier-Stokes equations, due to the fundamental architecture of the GPU hardware being unsuited for elliptic problems where communication patterns are non-local. If one were to apply the multigrid-lattice Boltzmann method running on GPUs as a solver for the pressure Poisson equation coming from the incompressible Navier-Stokes equations, one would likely obtain very good scaling and speedup results. This is perhaps the only way GPUs will be useful for incompressible flow solvers.

Other interesting recent developments in the level-set literature include the work by Della Rocca *et al.* [267] on boundary conditions

for the level-set function. In the code employed here, the simple choice of mirroring the level-set function at the boundaries is employed. For axisymmetry this is the correct choice, but for e.g. a drop sliding down a wall it will not give the correct behaviour. The method by della Rocca et al. gives correct behaviour for this case and others, and is not difficult to implement. Another interesting development is the previously mentioned work by Sabelnikov *et al.* [177], who present a new algorithm which replaces the need for velocity extrapolation before the level-set function is advected. This new algorithm has significantly lower computational cost than velocity extrapolation, and again it appears to be relatively straightforward to implement.

Towards the physical modelling end of the spectrum, there is also work to be done. In particular, modelling of soluble surfactants is an interesting topic. The transport, adsorption and desorption equations for soluble surfactants have been widely studied, and it is well-understood at this point how to implement this into a code such as that used here. An interesting application for soluble surfactants is to study the situation of a falling drop for diffusion-limited adsorption. In this case, it is known from the theory [81] that the drop will not be spherical. Simulations of this case would require a very large computational domain, and low Reynolds number, thus it is likely that such studies would require an implicit time integration method such as those discussed previously. Perhaps the best approach would be to solve the Stokes equation, rather than the Navier-Stokes equations. In any case, results which quantified the degree of non-sphericity in a significant region of the parameter space would surely be interesting.

When it comes to the theoretical results presented in Paper 5, further simulations could also further strengthen these. But perhaps the most interesting approach would be to use nuclear magnetic resonance velocity measurements, e.g. as in the work by Han *et al.* [268], to obtain directly the velocity field inside drops falling at velocity between the Stokes and the Hadamard-Rybzynski results. This would allow a direct test of the stagnant-cap model and of the continuous-interface model presented in Paper 5. It could also determine whether the tracer particles used for flow visualisation in experimental studies are the real cause of the observed stagnant cap, a hypothesis put forward in Paper 5.

With the hybrid level-set/ghost-fluid/immersed-boundary method developed here for Paper 8, an obvious next step is to switch from

[81] Levich, V. G. (1962)

the immersed-boundary to the immersed-interface method, since this would mean the overall method would be a sharp-interface method.

**Suggestions for the molecular simulations**

One issue to which a satisfactory solution should be found is the large cutoffs currently used in simulations. Since the amount of work grows with the cutoff to the third power, having as large a cutoff as possible would obviously be beneficial. A systematic study of the effect of the cutoff size in different systems (liquid-liquid, liquid-vapor, with a third dispersed phase, etc.) should be undertaken.

In a similar spirit, it would be a fruitful venture to construct an "autotuner" for the binary interaction parameter $k_{ij}$ used in cross-interactions. Using raaSAFT this would be relatively straight-forward, based on the combination of two things which must be added to the code: a built-in estimator for the physical property to which one is tuning (e.g. interfacial tension), and a Newton-Rhapson method for obtaining the next guess for the $k_{ij}$. Both the first in itself and the combination of both are useful and low-hanging fruit.

When it comes to Bottled SAFT, several topics are interesting for future investigations. One is simply to study the vast number of chemical compounds available in this database; as an example, several small molecule drugs like aspirin and ibuprofen have models in Bottled SAFT. Another topic for future work is to predict physicochemical properties, such as pure-component surface tension, directly from the model parameters without having to run a molecular dynamics simulation.

**Suggestions for investigations of the asphaltenes**

When it comes to the asphaltenes, there are of course also many things to be studied in the future. An interesting topic (suggested to the author by M. Hoepfner) is that of ergodicity in these systems, i.e. whether the time evolution of a system samples from the entire configuration space, both for asphaltenes in the bulk and at the interface. When studying ergodicity in the simulations, it is natural to also consider the topic of ergodicity breaking in the physical system.

As remarked several times already, truly taking polydispersity into account will likely fuel the next significant improvement in molecular models for asphaltenes. To accomplish this, there are several interesting approaches to consider. Regardless of which one chooses, it is highly likely that coarse-grained methods will continue

to hold a decisive advantage over atomistic simulations. Thus one is brought to consider the construction of a coarse-grained QMR approach, whereby one can obtain many (of the order of 10, perhaps even 100) different model asphaltene molecules representative of real-life experimental data.

A second approach would be enabled if analytical chemistry techniques keep improving at the current pace. If in the future one is able to actually measure the molecular structure of all the different asphaltene molecules in a given sample, putting these into a simulation would be a small challenge in comparison. Such a brute force approach would however require an extremely large simulation, and thus would also require runtime on the largest GPU-clusters in the world.

When considering such a brute force approach, an interesting recent paper by Jacobs *et al.*[269] comes to mind. In this theoretical approach, the self-assembly of many pieces, each one different from all the others, is considered. While the application in that paper is nano-engineering or DNA assembly, the mind is brought to consider how this approach could be applied to the topic of asphaltenes.

In addition to these matters, there is also the effect of the remaining components of crude oil to consider. The effect of crude oil composition, in particular the effect of resins, as well as the simulation of chemical demulsifiers, are all interesting topics for future study. As remarked in Paper 8, the parameter space for the crude oil/water system is discouragingly large, so one has to be very clever when undertaking such studies.

Furthermore, when it comes to the topic of electrocoalescence, the effect of the electric field on the asphaltene molecules at the interface has not previously been studied using molecular simulations. A very interesting question in this regard is how localised the change in direction of the electric field at the interface is on the nanoscale. If it is sufficiently localised, is the field strong enough to exert a significant force on the asphaltenes, which are known to contain polar groups? How does this change the interfacial behaviour of the asphaltenes? It should be noted that the electric field, varying typically with a frequency of 50 Hz, can be assumed to be constant at the molecular scale.

Having but limited experience with experimental work, we end this section with a more speculative suggestion. The atomic force microscopy studies presented by Schuler *et al.*[136], which show detailed images of individual asphaltene molecules such as the one reproduced in Figure 2.2, are extremely impressive. However, these

asphaltenes under study are isolated and then deposited on a metal surface, a situation very different from their natural environment. Imagine if, instead, one was to put an extremely low API (i.e. high density) crude oil in a box, put water over the crude, wait for some time, and then very quickly freeze the system. If the ice was subsequently sublimated away at a low pressure, one would be left with just the imprint of the crude oil interface in its natural state when it contacts water (if no extreme changes occur in the interface during freezing, which may of course not be true). Assuming this is physically realisable; if one was then to study the surface of this block of frozen oil using the same atomic force microscopy, what would one see? Would it be like looking at the crude oil interface from within the water at the molecular scale, as is done in the simulations presented in Paper 8? What does this interface really look like? One can but hope that future studies may be able to unravel these mysteries.

# Bibliography


1. SPEED, B. An Appreciation of Dr. Cottrell. *Industrial & Engineering Chemistry*, **11**, 153–154 (1919)

2. COTTRELL, F. G. & SPEED, J. B. *Separating and collecting particles of one liquid suspended in another liquid.* US Patent 987,115. 1911

3. BITTO, R. *et al. Special Centennial Issue. Baker Hughes InDepth* **13** (Baker Hughes, 2007)

4. GRANT, T. *International Directory of Company Histories* (St. James Press, 1996)

5. WATERMAN, L. C. Electrical coalescers. *Chemical Engineering Progress*, **61**, 51–55 (1965)

6. TAYLOR, G. Disintegration of Water Drops in an Electric Field. *Proceedings of the Royal Society of London A*, **280**, 383–397. doi:10.1098/rspa.1964.0151 (1964)

7. ZELENY, J. Instability of Electrified Liquid Surfaces. *Physical Review*, **10**, 1–6. doi:10.1103/PhysRev.10.1 (1917)

8. WILSON, C. T. R. & TAYLOR, G. I. The bursting of soap-bubbles in a uniform electric field. *Mathematical Proceedings of the Cambridge Philosophical Society*, **22**, 728–730. doi:10.1017/S0305004100009609 (1925)

9. MACKY, W. A. Some Investigations on the Deformation and Breaking of Water Drops in Strong Electric Fields. *Proceedings of the Royal Society of London A*, **133**, 565. doi:10.1098/rspa.1931.0168 (1931)

10. BRAZIER-SMITH, P. R. Stability and Shape of Isolated and Pairs of Water Drops in an Electric Field. *Physics of Fluids*, **14**, 1–6. doi:10.1063/1.1693258 (1971)

11. MELCHER, J. R. & TAYLOR, G. I. Electrohydrodynamics: a review of the role of interfacial shear stresses. *Annual Review of Fluid Mechanics*, **1**, 111–146. doi:10.1146/annurev.fl.01.010169.000551 (1969)

12. MOHAMMED, R., BAILEY, A., LUCKHAM, P. & TAYLOR, S. Dewatering of crude oil emulsions 1. Rheological behaviour of the crude oil—water interface. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, **80**, 223–235. doi:10.1016/0927-7757(93)80202-P (1993)

13. MOHAMMED, R., BAILEY, A., LUCKHAM, P. & TAYLOR, S. Dewatering of crude oil emulsions 2. Interfacial properties of the asphaltic constituents of crude oil. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, **80**, 237–242. doi:10.1016/0927-7757(93)80203-Q (1993)

14. MOHAMMED, R., BAILEY, A., LUCKHAM, P. & TAYLOR, S. Dewatering of crude oil emulsions 3. Emulsion resolution by chemical means. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, **83**, 261–271. doi:10.1016/0927-7757(93)02706-K (1994)

15. CHEN, T., MOHAMMED, R., BAILEY, A., LUCKHAM, P. & TAYLOR, S. Dewatering of crude oil emulsions 4. Emulsion resolution by the application of an electric field. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, **83**, 273–284. doi:10.1016/0927-7757(93)02653-V (1994)

16. JOHANSEN, E. J., SKJÄRVÖ, I. M., LUND, T., SJÖBLOM, J., SÖDERLUND, H. & BOSTRÖM, G. Water-in-crude oil emulsions from the Norwegian Continental Shelf 1. Formation, characterization and stability correlations. *Colloids and Surfaces*, **34**, 353–370. doi:10.1016/0166-6622(88)80160-4 (1989)

17. SJÖBLOM, J., MINGYUAN, L., HÖILAND, H. & JOHANSEN, E. J. Water-in-crude oil emulsions from the Norwegian Continental Shelf 2. Chemical destabilization and interfacial tensions. *Colloid and Polymer Science*, **268**, 389–398. doi:10.1007/BF01411682 (1990)

18. SJÖBLOM, J., MINGYUAN, L., HÖILAND, H. & JOHANSEN, E. J. Water-in-crude oil emulsions from the Norwegian Continental Shelf 3. A comparative destabilization of model systems. *Colloids and Surfaces*, **46**, 127–139. doi:10.1016/0166-6622(90)80161-V (1990)

19. NORDLI, K. G., SJÖBLOM, J., KIZLING, J. & STENIUS, P. Water-in-crude oil emulsions from the Norwegian Continental Shelf 4. Monolayer properties of the interfacially active crude oil fraction. *Colloids and Surfaces*, **57**, 83–98. doi:10.1016/0166-6622(91)80182-N (1991)

20. BØRVE, K. G. N., SJÖBLOM, J. & STENIUS, P. Water-in-crude oil emulsions from the Norwegian Continental Shelf 5. A comparative monolayer study of model polymers. *Colloids and Surfaces*, **63**, 241–251. doi:10.1016/0166-6622(92)80245-W (1992)

21. MINGYUAN, L., CHRISTY, A. A. & SJØBLOM, J. in *Emulsions—A Fundamental and Practical Approach* 157–172 (Springer, 1992)

22. SJÖBLOM, J., MINGYUAN, L., CHRISTY, A. A. & GU, T. Water-in-crude-oil emulsions from the Norwegian Continental Shelf 7. Interfacial pressure and emulsion stability. *Colloids and Surfaces*, **66**, 55–62. doi:10.1016/0166-6622(92)80120-Q (1992)

23. URDAHL, O., SJÖBLOM, J., *et al.* Water-in-crude oil emulsions from the Norwegian Continental Shelf 8. Surfactant and macromolecular destabilization. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, **74**, 293–302. doi:10.1016/0927-7757(93)80273-H (1993)

24. SKODVIN, T., SJÖBLOM, J., SAETEN, J. O., URDAHL, O. & GESTBLOM, B. Water-in-crude oil emulsions from the Norwegian Continental Shelf 9. A dielectric spectroscopic characterization of authentic as well as model systems. *Journal of Colloid and Interface Science*, **166**, 43–50. doi:10.1006/jcis.1994.1269 (1994)

25. SJÖBLOM, J., MINGYUAN, L., CHRISTY, A. A. & RØNNINGSEN, H. P. Water-in-crude oil emulsions from the Norwegian Continental Shelf 10. Ageing of the interfacially active components and the influence on the emulsion stability. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, **96**, 261–272. doi:10.1016/0927-7757(94)03038-2 (1995)

26. RØNNINGSEN, H. P., SJÖBLOM, J. & MINGYUAN, L. Water-in-crude oil emulsions from the Norwegian Continental Shelf 11. Ageing of crude oils and its influence on the emulsion stability. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, **97**, 119–128. doi:10.1016/0927-7757(94)03039-3 (1995)

27. URDAHL, O. & SJÖBLOM, J. Water-in-crude oil emulsions from the Norwegian Continental Shelf 12. A stabilization and destabilization study. *Journal of Dispersion Science and Technology*, **16**, 557–574. doi:10.1080/01932699508943707 (1995)

28. CHIESA, M., MELHEIM, J., PEDERSEN, A., INGEBRIGTSEN, S. & BERG, G. Forces acting on water droplets falling in oil under the influence of an electric field: numerical predictions versus experimental observations. *European Journal of Mechanics - B/Fluids*, **24**, 717–732. doi:10.1016/j.euromechflu.2005.03.005 (2005)

29. MELHEIM, J. A. & CHIESA, M. Simulation of turbulent electrocoalescence. *Chemical Engineering Science*, **61**, 4540–4549. doi:10.1016/j.ces.2006.02.022 (2006)

30. HANSEN, E. *Numerical simulation of droplet dynamics in the presence of an electric field.* PhD thesis (NTNU, 2005)

31. BJØRKLUND, E. The level-set method applied to droplet dynamics in the presence of an electric field. *Computers & Fluids*, **38**, 358–369. doi:10.1016/j.compfluid.2008.04.008 (2009)

32. HELLESØ, S., ATTEN, P., BERG, G. & LUNDGAARD, L. Experimental study of electrocoalescence of water drops in crude oil using near-infrared camera. *Experiments in Fluids*, **56**. doi:10.1007/s00348-015-1990-y (2015)

33. ATTEN, P., LUNDGAARD, L. & BERG, G. A simplified model of electrocoalescence of two close water droplets in oil. *Journal of Electrostatics*, **64**. Fifth International Electrohydrodynamics (EHD) Workshop and Fourth Conference of the Société Française d'Electrostatique (SFE), 550–554. doi:10.1016/j.elstat.2005.10.009 (2006)

34. LUNDGAARD, L. E., BERG, G., INGEBRIGTSEN, S. & ATTEN, P. in *Emulsions and Emulsion Stability* (ed SJÖBLOM, J.) 549–592 (Taylor & Francis, 2006)

35. TEIGEN, K. E. & MUNKEJORD, S. T. Sharp-interface simulations of drop deformation in electric fields. *IEEE Transactions on Dielectric and Electrical Insulation*, **16**, 475. doi:10.1109/TDEI.2009.4815181 (2009)

36. TEIGEN, K. E. & MUNKEJORD, S. T. Influence of surfactant on drop deformation in an electric field. *Physics of Fluids*, **22**, 112104. doi:10.1063/1.3504271 (2010)

37. TEIGEN, K. E., SONG, P., LOWENGRUB, J. & VOIGT, A. A diffuse-interface method for two-phase flows with soluble surfactants. *Journal of Computational Physics*, **230**, 375–393. doi:10.1016/j.jcp.2010.09.020 (2011)

38. TEIGEN, K. *Development and use of interface-capturing methods for investigation of surfactant-covered drops in electric fields.* PhD thesis (NTNU, 2010)

39. JONES, T., NEUSTADTER, E., WHITTINGHAM, K., *et al.* Water-in-crude oil emulsion stability and emulsion destabilization by chemical demulsifiers. *Journal of Canadian Petroleum Technology*, **17**. doi:10.2118/78-02-08 (1978)

40. GAFONOVA, O. V. & YARRANTON, H. W. The stabilization of water-in-hydrocarbon emulsions by asphaltenes and resins. *Journal of Colloid and Interface Science*, **241**, 469–478. doi:10.1006/jcis.2001.7731 (2001)

41. SJÖBLOM, J. *Encyclopedic handbook of emulsion technology* (CRC Press, 2001)

42. KOKAL, S. L. Crude Oil Emulsions: A State-Of-The-Art Review. *SPE Production & Facilities*, **20**, 5–13. doi:10.2118/77497-PA (2005)

43. KILPATRICK, P. K. Water-in-Crude Oil Emulsion Stabilization: Review and Unanswered Questions. *Energy & Fuels*, **26**, 4017–4026. doi:10.1021/ef3003262 (2012)

44. KOLAHDOUZ, E. M. *Modeling the electrohydrodynamics of three-dimensional vesicles.* PhD thesis (University at Buffalo SUNY, 2015)

45. TIELENS, A. G. Interstellar polycyclic aromatic hydrocarbon molecules. *Annual Review of Astronomy and Astrophysics*, **46**, 289–337. doi:10.1146/annurev.astro.46.060407.145211 (2008)

46. BERNSTEIN, M. P., SANDFORD, S. A., ALLAMANDOLA, L. J., GILLETTE, J. S., CLEMETT, S. J. & ZARE, R. N. UV Irradiation of Polycyclic Aromatic Hydrocarbons in Ices: Production of Alcohols, Quinones, and Ethers. *Science*, **283**, 1135–1138. doi:10.1126/science.283.5405.1135 (1999)

47. BERNOULLI, D. *Hydrodynamica* (Johann Reinhold Dulsecker, 1738)

48. BERNOULLI, D. *Die Werke von Daniel Bernoulli. Band 5: Hydrodynamik II* (eds MIKHAILOV, G. K., RADELET-DE GRAVE, P. & SPEISER, D.) (Birkhäuser, 2002)

49. BERNOULLI, D. & BERNOULLI, J. *Hydrodynamics, by Daniel Bernoulli, and Hydraulics, by Johann Bernoulli.* (eds CARMODY, T. & KOBUS, H.) (Dover, 1968)

50. BINNIE, A. M. & EASTERLING, H. J. Hydrodynamics, by Daniel Bernoulli, and Hydraulics, by Johann Bernoulli. Translated from the Latin by T. Carmody and H. Kobus. Dover, 1968. *Journal of Fluid Mechanics*, **38**, 855–856. doi:10.1017/S0022112069212655 (1969)

51. MAXWELL, J. C. The Bakerian lecture: on the viscosity or internal friction of air and other gases. *Philosophical Transactions of the Royal Society of London*, **156**, 249–268. doi:10.1098/rspl.1866.0009 (1866)

52. MAXWELL, J. C. On the dynamical theory of gases. *Philosophical Transactions of the Royal Society of London*, **157**, 49–88. doi:10.1098/rstl.1867.0004 (1867)

53. VINCE, S. The Bakerian Lecture, 1797. Experiments upon the Resistance of Bodies Moving in Fluids. *Philosophical Transactions of the Royal Society of London*, **88**, 1–14. doi:10.1098/rstl.1798.0002 (1798)

54. BERNAL, J. D. The Bakerian Lecture, 1962. The Structure of Liquids. *Proceedings of the Royal Society of London A*, **280**, 299–322. doi:10.1098/rspa.1964.0147 (1964)

55. NAGATA, Y., USUI, K. & BONN, M. Molecular Mechanism of Water Evaporation. *Physical Review Letters*, **115**, 236102. doi:10.1103/PhysRevLett.115.236102 (2015)

56. KNUDSEN, M. *The kinetic theory of gases. Some modern aspects.* (Methuen, 1934)

57. PAN, Y. & SUGA, K. Numerical simulation of binary liquid droplet collision. *Physics of Fluids*, **17**, 082105. doi:10.1063/1.2009527 (2005)

58. QIAN, J. & LAW, C. Regimes of coalescence and separation in droplet collision. *Journal of Fluid Mechanics*, **331**, 59–80. doi:10.1017/S0022112096003722 (1997)

59. LAU, G. V., FORD, I. J., HUNT, P. A., MÜLLER, E. A. & JACKSON, G. Surface thermodynamics of planar, cylindrical, and spherical vapour-liquid interfaces of water. *The Journal of Chemical Physics*, **142**. doi:10.1063/1.4913371 (2015)

60. SUPPLE, S. & QUIRKE, N. Rapid Imbibition of Fluids in Carbon Nanotubes. *Physical Review Letters*, **90**, 214501. doi:10.1103/PhysRevLett.90.214501 (May 2003)

61. DIMITROV, D. I., MILCHEV, A. & BINDER, K. Capillary Rise in Nanopores: Molecular Dynamics Evidence for the Lucas-Washburn Equation. *Physical Review Letters*, **99**, 054501. doi:10.1103/PhysRevLett.99.054501 (2007)

62. LUCAS, R. Über das Zeitgesetz des kapillaren Aufstiegs von Flüssigkeiten. *Kolloid-Zeitschrift*, **23**, 15–22. doi:10.1007/BF01461107 (1918)

63. WASHBURN, E. W. The Dynamics of Capillary Flow. *Physical Review*, **17**, 273–283. doi:10.1103/PhysRev.17.273 (Mar. 1921)

64. GAD-EL-HAK, M. Gas and Liquid Transport at the Microscale. *Heat Transfer Engineering*, **27**, 13–29. doi:10.1080/01457630500522305 (2006)

65. CHAPMAN, S., COWLING, T. G. & CERCIGNANI, C. *The Mathematical Theory of Non-uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases* (Cambridge Mathematical Library, 1991)

66. LIOUVILLE, J. Note sur la Théorie de la Variation des constantes arbitraires. *Journal de Mathématiques Pures et Appliquées*, **3**, 342–349 (1838)

67. BOGOLIUBOV, N. N. Kinetic Equations. *Journal of Physics USSR*, **10**, 265–274 (1946)

68. YVON, J. *La théorie statistique des fluides et l'équation d'état* (Hermann & cie, 1935)

69. KIRKWOOD, J. G. The Statistical Mechanical Theory of Transport Processes I. General Theory. *The Journal of Chemical Physics*, **14**, 180. doi:10.1063/1.1724117 (1946)

70. BORN, M. & GREEN, H. S. A General Kinetic Theory of Liquids I. The Molecular Distribution Functions. *Proceedings of the Royal Society of London A*, **188**, 10–18. doi:10.1098/rspa.1946.0093 (1946)

71. BATCHELOR, G. *An Introduction to Fluid Dynamics* (Cambridge University Press, 2000)

72. AGARWAL, R. K., YUN, K.-Y. & BALAKRISHNAN, R. Beyond Navier–Stokes: Burnett equations for flows in the continuum–transition regime. *Physics of Fluids*, **13**, 3061–3085. doi:10.1063/1.1397256 (2001)

73. ZHONG, X., MACCORMACK, R. W. & CHAPMAN, D. R. Stabilization of the Burnett equations and application to hypersonic flows. *AIAA Journal*, **31**, 1036–1043 (1993)

74. JIN, S. & SLEMROD, M. Regularization of the Burnett equations via relaxation. *Journal of Statistical Physics*, **103**, 1009–1033. doi:10.1023/A:1010365123288 (2001)

75. STRUCHTRUP, H. Failures of the Burnett and super-Burnett equations in steady state processes. *Continuum Mechanics and Thermodynamics*, **17**, 43–50. doi:10.1007/s00161-004-0186-0 (2005)

76. SÖDERHOLM, L. H. Hybrid Burnett equations: A new method of stabilizing. *Transport Theory and Statistical Physics*, **36**, 495–512. doi:10.1080/00411450701468365 (2007)

77. COMEAUX, K. A., CHAPMAN, D. R. & MACCORMACK, R. W. *An analysis of the Burnett equations based on the second law of thermodynamics* in *33. AIAA Aerospace Sciences Meeting and Exhibit* (1995)

78. BHATNAGAR, P. L., GROSS, E. P. & KROOK, M. A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Physical Review*, **94**, 511–525. doi:10.1103/PhysRev.94.511 (1954)

79. PURCELL, E. M. Life at low Reynolds number. *American Journal of Physics*, **45**, 3–11 (1977)

80. CLIFT, R., GRACE, J. & WEBER, E. *Bubbles, Drops, and Particles* (Dover Publications, 2013)

81. LEVICH, V. G. *Physicochemical hydrodynamics* 2nd (Prentice-Hall, 1962)

82. LEAL, L. G. *Advanced transport phenomena: Fluid mechanics and convective transport processes* (Cambridge University Press, 2007)

83. LEVY, S. *Two-phase flow in complex systems* (John Wiley & Sons, 1999)

84. BRAUNER, N. *Liquid-liquid two-phase flow systems* (Springer, 2003)

85. BIDONE, G. Expériences sur la forme et sur la direction des veines et des courants d'eau lancés par diverses ouvertures. *Memorie della Reale Accademia delle scienze di Torino*, 229 (1829)

86. RAYLEIGH, L. On the capillary phenomena of jets. *Proceedings of the Royal Society of London*, **29**, 71–97. doi:10.1098/rspl.1879.0015 (1879)

87. LAMB, H. *Hydrodynamics* 6th ed., 599 (Dover, New York, 1945)

88. CHANDRASEKHAR, S. The Oscillations of a Viscous Liquid Globe. *Proceedings of the London Mathematical Society*, **s3-9**, 141–149. doi:10.1112/plms/s3-9.1.141 (1959)

89. MILLER, C. & SCRIVEN, L. The oscillations of a fluid droplet immersed in another fluid. *Journal of Fluid Mechanics*, **32**, 417–435. doi:10.1017/S0022112068000832 (1968)

90. PROSPERETTI, A. Free oscillations of drops and bubbles: the initial-value problem. *Journal of Fluid Mechanics*, **100**, 333–347. doi:10.1017/S0022112080001188 (1980)

91. RONAY, M. Determination of the Dynamic Surface Tension of Liquids from the Instability of Excited Capillary Jets and from the Oscillation Frequency of Drops Issued from Such Jets. *Proceedings of the Royal Society of London A*, **361**, 181–206. doi:10.1098/rspa.1978.0098 (1978)

92. HADAMARD, J. Mouvement permanent lent d'une sphère liquide et visqueuse dans un liquide visqueux. *Comptes Rendus de l'Académie des Sciences*, **152**, 1735–1738 (1911)

93. RYBZYNSKI, W. Über die fortschreitende Bewegung einer flüssigen Kugel in einem zähen Medium. *Bulletin International de l'Académie des Sciences de Cracovie*, (1911)

94. STOKES, G. G. On the Effect of the Internal Friction of Fluids on the Motion of Pendulums. *Transactions of the Cambridge Philosophical Society*, **9**, 8 (1851)

95. NORDLUND, I. The Validity of Stokes' Law for the Motion of Liquid Drops in other Liquids. *Arkiv för matematik, astronomi och fysik*, **9** (1913)

96. LEBEDEV, A. Stokes' law as applied to liquid balls (In Russian). *Zhurnal Fizicheskoi Khimii*, **48** (1916)

97. SILVEY, O. W. The Fall of Mercury Droplets in a Viscous Medium. *Physical Review*, **7**, 106–111. doi:10.1103/PhysRev.7.106 (1916)

98. BOND, W. N. LXXXII. Bubbles and drops and Stokes' law. *Philosophical Magazine*, **4**, 889–898. doi:10.1080/14786441108564394 (1927)

99. BOND, W. N. & NEWTON, D. A. LXXXII. Bubbles and drops and Stokes' law. (Paper 2). *Philosophical Magazine*, **5**, 794–800. doi:10.1080/14786440408564523 (1928)

100. BOUSSINESQ, J. Sur l'existence d'une viscosité superficielle, dans la mince couche de transition séparant un liquide d'un autre fluide contigu. *Comptes Rendus de l'Académie des Sciences*, **156**, 983–989 (1913)

101. MARANGONI, C. *Sull'espansione Delle Gocce di un Liquido Gallegianti Sulla Superficie di Altro Liquido.* PhD thesis (University of Pavia, 1865)

102. MARANGONI, C. Über die Ausbreitung der Tropfen einer Flussigkeit auf der Oberfläche einer anderen. *Annalen der Physik und Chemie*, **143**, 337–354. doi:10.1002/andp.18712190702 (1870)

103. SAKATA, E. & BERG, J. Surface diffusion in monolayers. *Industrial & Engineering Chemistry Fundamentals*, **8**, 570–575. doi:10.1021/i160031a033 (1969)

104. VOLAROVICH, M. & LEONT'YEVA, A. Unknown title (In Russian), cited in Levich: "Physicochemical Hydrodynamics". *Acta Physiochimica URSS*, **11**, 251 (1939)

105. FRUMKIN, A. & BAGOTSKAYA, I. Unknown title (In Russian), cited in Levich: "Physicochemical Hydrodynamics". *Comptes Rendus de l'Académie des Sciences de l'URSS*, **55** (1947)

106. FRUMKIN, A. & LEVICH, V. Effect of surface-active substances on movements at the boundaries of liquid phases. *Zhurnal Fizicheskoi Khimii*, **21**, 1183–1204 (1947)

107. DUKHIN, S. S., RULYOV, N. N. & DIMITROV, D. S. *Coagulation and Dynamics of Thin Films (in Russian)* (Naukova Dumka, 1986)

108. LEVICH, V. & KRYLOV, V. Surface-tension-driven phenomena. *Annual Review of Fluid Mechanics*, **1**, 293–316. doi:10.1146/annurev.fl.01.010169.001453 (1969)

109. PLINY THE ELDER. *Naturalis Historia* (ed MAYHOFF, K. C.) http://penelope.uchicago.edu/Thayer/E/Roman/Texts/Pliny_the_Elder/home.html (Teubner, Leipzig, 1906)

110. PLINY THE ELDER. *Pliny's Natural History* (ed RACKHAM, H.) http://www.masseiana.org/pliny.htm (Harvard University Press, Massachusetts, 1949)

111. FRANKLIN, B. Of the stilling of waves by means of oil. *Philosophical Transactions of the Royal Society of London*, **64**, 445–60 (1774)

112. TANFORD, C. *Ben Franklin Stilled the Waves* (Oxford University Press, 2004)

113. RAYLEIGH, L. Measurements of the amount of oil necessary in order to check the motions of camphor upon water. *Proceedings of the Royal Society of London A*, **47**, 364–367. doi:10.1098/rspl.1889.0099 (1889)

114. RAYLEIGH, L. On the superficial viscosity of water. *Proceedings of the Royal Society of London A*, **48**, 127–140. doi:10.1098/rspl.1890.0018 (1890)

115. POCKELS, A. Surface tension. *Nature*, **43**, 437. doi:10.1038/043437c0 (1891)

116. PLATEAU, J. *Statique Expérimentale et Théorique des Liquides Soumis aux Seules Forces Moléculaires* (Gauthier-Villars, Paris, 1873)

117. MARANGONI, C. The principle of the surface viscosity of liquids established by Mr. J. Plateau. *Nuovo Cimento*, **2**, 239–273 (1872)

118. POCKELS, A. Relations Between the Surface-Tension and Relative Contamination of Water Surfaces. *Nature*, **48**, 152–154. doi:10.1038/048152a0 (1893)

119. POCKELS, A. The measurement of surface tension with the balance. *Science*, **64**, 304. doi:10.1126/science.64.1656.304 (1926)

120. OSTWALD, W. Die Arbeiten von Agnes Pockels über Grenzschichten und Filme. *Kolloid-Zeitscrift*, **58**, 1–8 (1932)

121. ROBERTS, G. *Langmuir-Blodgett films* (Springer Science & Business Media, 2013)

122. VAN DEN TEMPEL, M. *The Function of Stabilizers during Emulsification* in *Proceedings of the 3rd International Congress on Surface Active Agents* **2** (Cologne, Germany, 1960), 573

123. LUCASSEN-REYNDERS, E., CAGNA, A. & LUCASSEN, J. Gibbs elasticity, surface dilational modulus and diffusional relaxation in nonionic surfactant monolayers. *Colloids and Surfaces A*, **186**, 63–72. doi:http://dx.doi.org/10.1016/S0927-7757(01)00483-6 (2001)

124. LUCASSEN, J. Longitudinal capillary waves. Part 1. Theory. *Transactions of the Faraday Society*, **64**, 2221–2229. doi:10.1039/TF9686402221 (1968)

125. GIBBS, J. W. On the equlibrium of heterogeneous substances. *Transactions of the Connecticut Academy of Arts and Sciences*, **143**, 337–354 (1878)

126. FULLER, G. G. & VERMANT, J. Complex Fluid-Fluid Interfaces: Rheology and Structure. *Annual Review of Chemical and Biomolecular Engineering*, **3**, 519–543. doi:10.1146/annurev-chembioeng-061010-114202 (2012)

127. LUCASSEN-REYNDERS, E. Dynamic interfacial properties in emulsification. *Encyclopedia of Emulsion Technology*, **4**, 63–91 (1996)

128. BOUSSINGAULT, M. XCI. On the constitution of bitumens. *Philosophical Magazine*, **9**, 487–488. doi:10.1080/14786443608649048 (1836)

129. HOEPFNER, M. P. *Investigations into Asphaltene Deposition, Stability, and Structure*. PhD thesis (University of Michigan, 2013)

130. DICKIE, J. P. & YEN, T. F. Macrostrucutres of asphaltic fractions by various instrumental methods. *Analytical Chemistry*, **39**, 1847–1852. doi:10.1021/ac50157a057 (1967)

131. MULLINS, O. C. The modified Yen model. *Energy & Fuels*, **24**, 2179–2207. doi:10.1021/ef900975e (2010)

132. MULLINS, O. C. The asphaltenes. *Annual Review of Analytical Chemistry*, **4**, 393–418. doi:10.1146/annurev-anchem-061010-113849 (2011)

133. MULLINS, O. C., SABBAH, H., EYSSAUTIER, J., POMERANTZ, A. E., BARRÉ, L., ANDREWS, A. B., RUIZ-MORALES, Y., MOSTOWFI, F., MCFARLANE, R., GOUAL, L., *et al.* Advances in asphaltene science and the Yen–Mullins model. *Energy & Fuels*, **26**, 3986–4003. doi:10.1021/ef300185p (2012)

134. EVDOKIMOV, I. N., FESAN, A. A. & LOSEV, A. P. New Answers to the Optical Interrogation of Asphaltenes: Monomers and Primary Aggregates from Steady-State Fluorescence Studies. *Energy & Fuels*, Article In Press. doi:10.1021/acs.energyfuels.6b00027 (2016)

135. RODGERS, R. P., SCHAUB, T. M. & MARSHALL, A. G. Petroleomics: MS Returns to Its Roots. *Analytical Chemistry*, **77**, 20–27. doi:10.1021/ac053302y (2005)

136. SCHULER, B., MEYER, G., PEÑA, D., MULLINS, O. C. & GROSS, L. Unraveling the Molecular Structures of Asphaltenes by Atomic Force Microscopy. *Journal of the American Chemical Society*, **137**, 9870–9876. doi:10.1021/jacs.5b04056 (2015)

137. GOUAL, L. & FIROOZABADI, A. Measuring asphaltenes and resins, and dipole moment in petroleum fluids. *AIChE Journal*, **48**, 2646–2663 (2002)

138. MCKENNA, A. M., MARSHALL, A. G. & RODGERS, R. P. Heavy Petroleum Composition. 4. Asphaltene Compositional Space. *Energy & Fuels*, **27**, 1257–1267. doi:10.1021/ef301747d (2013)

139. PODGORSKI, D. C., CORILO, Y. E., NYADONG, L., LOBODIN, V. V., BYTHELL, B. J., ROBBINS, W. K., MCKENNA, A. M., MARSHALL, A. G. & RODGERS, R. P. Heavy Petroleum Composition. 5. Compositional and Structural Continuum of Petroleum Revealed. *Energy & Fuels*, **27**, 1268–1276. doi:10.1021/ef301737f (2013)

140. TANG, W., HURT, M. R., SHENG, H., RIEDEMAN, J. S., BORTON, D. J., SLATER, P. & KENTTÄMAA, H. I. Structural Comparison of Asphaltenes of Different Origins Using Multi-stage Tandem Mass Spectrometry. *Energy & Fuels*, **29**, 1309–1314. doi:10.1021/ef501242k (2015)

141. YEUNG, A., DABROS, T., CZARNECKI, J. & MASLIYAH, J. On the interfacial properties of micrometre–sized water droplets in crude oil. *Proceedings of the Royal Society of London A*, **455**, 3709–3723. doi:10.1098/rspa.1999.0473 (1999)

142. JIANG, G.-S. & SHU, C.-W. Efficient Implementation of Weighted ENO Schemes. *Journal of Computational Physics*, **126**, 202–228. doi:10.1006/jcph.1996.0130 (1996)

143. JIANG, G.-S. & PENG, D. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM Journal on Scientific Computing*, **21**, 2126–2143. doi:10.1137/S106482759732455X (2000)

144. RHIE, C. M. & CHOW, W. L. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, **21**, 1525–1532. doi:10.2514/3.8284 (1983)

145. CHORIN, A. J. Numerical solution of the Navier-Stokes equations. *Mathematics of computation*, **22**, 745–762. doi:10.1090/S0025-5718-1968-0242392-2 (1968)

146. BHATIA, H., NORGARD, G., PASCUCCI, V. & BREMER, P.-T. The Helmholtz-Hodge Decomposition – A Survey. *Visualization and Computer Graphics, IEEE Transactions on*, **19**, 1386–1404. doi:10.1109/TVCG.2012.316 (2013)

147. DUFFY, A., KUHNLE, A. & SUSSMAN, M. An improved variable density pressure projection solver for adaptive meshes. Unpublished. Available online (2002)

148. VAN DER VORST, H. Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *SIAM Journal on Scientific Computing*, **13**, 631–644. doi:10.1137/0913035 (1992)

149. HENSON, V. E. & YANG, U. M. BoomerAMG: a Parallel Algebraic Multigrid Solver and Preconditioner. *Applied Numerical Mathematics*, **41**, 155–177 (2000)

150. FALGOUT, R., JONES, J. & YANG, U. in *Numerical Solution of Partial Differential Equations on Parallel Computers* 267–294 (Springer-Verlag, 2006)

151. BALAY, S., GROPP, W. D., MCINNES, L. C. & SMITH, B. F. in *Modern Software Tools for Scientific Computing* (eds BRUASET, A., ARGE, E. & LANGTANGEN, H.) 163–202 (Springer, 1997)

152. BALAY, S., BROWN, J., BUSCHELMAN, K., GROPP, W. D., KAUSHIK, D., KNEPLEY, M. G., MCINNES, L. C., SMITH, B. F. & ZHANG, H. *PETSc Web page* http://www.mcs.anl.gov/petsc. 2012

153. ZHUANG, Y. & SUN, X.-H. A High-Order Fast Direct Solver for Singular Poisson Equations. *Journal of Computational Physics*, **171**, 79–94. doi:10.1006/jcph.2001.6771 (2001)

154. YANG, J. *PETSc KSP tutorial ex43* https://www.mcs.anl.gov/petsc/petsc-current/src/ksp/ksp/examples/tutorials/ex34.c.html. 2012

155. KETCHESON, D. I. & ROBINSON, A. C. On the practical importance of the SSP property for Runge–Kutta time integrators for some common Godunov-type schemes. *International Journal for Numerical Methods in Fluids*, **48**, 271–303. doi:10.1002/fld.837 (2005)

156. GODUNOV, S. K. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, **89**, 271–306 (1959)

157. KANG, M., FEDKIW, R. & LIU, X.-D. A Boundary Condition Capturing Method for Multiphase Incompressible Flow. *Journal of Scientific Computing*, **15**, 323–360. doi:10.1023/A:1011178417620 (2000)

158. VON NEUMANN, J. & GOLDSTINE, H. H. Numerical inverting of matrices of high order. *Bulletin of the American Mathematical Society*, **53**, 1021–1099. doi:10.1090/S0002-9904-1947-08909-6 (1947)

159. GOLDBERG, D. What Every Computer Scientist Should Know About Floating Point Arithmetic. *ACM Computing Surveys*, **23**, 5–48. doi:10.1145/103162.103163 (1991)

160. JOHN, V., MATTHIES, G. & RANG, J. A comparison of time-discretization/linearization approaches for the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, **195**, 5995–6010. doi:10.1016/j.cma.2005.10.007 (2006)

161. GUERMOND, J.-L., MINEV, P. & SHEN, J. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, **195**, 6011–6045. doi:10.1016/j.cma.2005.10.010 (2006)

162. PIJL, S., SEGAL, A., VUIK, C. & WESSELING, P. Computing three-dimensional two-phase flows with a mass-conserving level set method. *Computing and Visualization in Science*, **11**, 221–235. doi:10.1007/s00791-008-0106-0 (2008)

163. VAN KAN, J. J. A second-order accurate pressure-correction scheme for viscous incompressible flow. *SIAM Journal on Scientific and Statistical Computing*, **7**, 870–891. doi:10.1137/0907059 (1986)

164. JOHNSTON, H. & LIU, J.-G. Accurate, stable and efficient Navier-Stokes solvers based on explicit treatment of the pressure term. *Journal of Computational Physics*, **199**, 221–259. doi:10.1016/j.jcp.2004.02.009 (2004)

165. SHEN, J. On Error Estimates of the Projection Methods for the Navier-Stokes Equations: Second-Order Schemes. *Mathematics of Computation*, **65**, 1039–1065. doi:10.1137/0729004 (1996)

166. LYSGAARD, M. O. *Implementation of an explicit incremental projection method for the Navier-Stokes equations* tech. rep. (Department of Mathematics, Norwegian University of Science and Technology, 2014)

167. HIRT, C. W. & NICHOLS, B. D. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, **39**, 201–225. doi:10.1016/0021-9991(81)90145-5 (1981)

168. OSHER, S. & SETHIAN, J. A. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, **79**, 12–49. doi:10.1016/0021-9991(88)90002-2 (1988)

169. UNVERDI, S. O. & TRYGGVASON, G. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, **100**, 25–37. doi:10.1016/0021-9991(92)90307-K (1992)

170. KARMA, A. & RAPPEL, W.-J. Phase-field method for computationally efficient modeling of solidification with arbitrary interface kinetics. *Physical Review E*, **53**, R3017. doi:10.1103/PhysRevE.53.R3017 (1996)

171. OSHER, S. & FEDKIW, R. *Level Set Methods and Dynamic Implicit Surfaces* (Springer New York, 2012)

172. SMEREKA, P. Semi-Implicit Level Set Methods for Curvature and Surface Diffusion Motion. *Journal of Scientific Computing*, **19**, 439–456. doi:10.1023/A:1025324613450 (2003)

173. ERVIK, Å., LERVÅG, K. Y. & MUNKEJORD, S. T. A robust method for calculating interface curvature and normal vectors using an extracted local level set. *Journal of Computational Physics*, **257**, 259–277. doi:10.1016/j.jcp.2013.09.053 (2014)

174. HARTMANN, D., MEINKE, M. & SCHRÖDER, W. The constrained reinitialization equation for level set methods. *Journal of Computational Physics*, **229**, 1514–1535. doi:DOI:10.1016/j.jcp.2009.10.042 (2010)

175. ERVIK, Å. Comment on "Numerical Study on Mobilization of Oil Slugs in Capillary Model with Level Set Approach", and Some Remarks on the Level Set Method. *Submitted to Engineering Applications of Computational Fluid Mechanics*, (2015)

176. ADALSTEINSSON, D. & SETHIAN, J. A. The Fast Construction of Extension Velocities in Level Set Methods. *Journal of Computational Physics*, **148**, 2–22. doi:10.1006/jcph.1998.6090 (1999)

177. SABELNIKOV, V., OVSYANNIKOV, A. Y. & GOROKHOVSKI, M. Modified level set equation and its numerical assessment. *Journal of Computational Physics*, **278**, 1–30. doi:10.1016/j.jcp.2014.08.018 (2014)

178. FEDKIW, R. P., ASLAM, T., MERRIMAN, B. & OSHER, S. A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method). *Journal of Computational Physics*, **152**, 457–492. doi:10.1006/jcph.1999.6236 (1999)

179. BRACKBILL, J., KOTHE, D. B. & ZEMACH, C. A continuum method for modeling surface tension. *Journal of Computational Physics*, **100**, 335–354. doi:10.1016/0021-9991(92)90240-Y (1992)

180. LERVÅG, K. Y. *Calculation of interface curvatures with the level-set method for two-phase flow simulations and a second-order diffuse-domain method for elliptic problems in complex geometries.* PhD thesis (Norwegian University of Science et al., 2013)

181. ANGOT, P., BRUNEAU, C.-H. & FABRIE, P. A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, **81**, 497–520. doi:10.1007/s002110050401 (1999)

182. PESKIN, C. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, **25**. cited By 1276, 220–252. doi:10.1016/0021-9991(77)90100-0 (1977)

183. PESKIN, C. & MCQUEEN, D. *A general method for the computer simulation of biological systems interacting with fluids* in *Symposia of the society for Experimental Biology* **49.** http://www.math.nyu.edu/~mcqueen/Public/papers/seb/SEB_19971216/SEB_19971216.html (1995), 265–276

184. FAI, T. G., GRIFFITH, B. E., MORI, Y. & PESKIN, C. S. Immersed Boundary Method for Variable Viscosity and Variable Density Problems Using Fast Constant-Coefficient Linear Solvers I: Numerical Method and Results. *SIAM Journal on Scientific Computing*, **35**, B1132–B1161. doi:10.1137/120903038 (2013)

185. LYSGAARD, M. O. *Development of an immersed boundary method for simulating contaminated fluid interfaces in two-phase flow.* MA thesis (Department of Mathematics, Norwegian University of Science and Technology, 2015)

186. KNOCHE, S. & KIERFELD, J. Buckling of spherical capsules. *Physical Review E*, **84**, 046608. doi:10.1103/PhysRevE.84.046608 (Oct. 2011)

187. KNOCHE, S., VELLA, D., AUMAITRE, E., DEGEN, P., REHAGE, H., CICUTA, P. & KIERFELD, J. Elastometry of Deflated Capsules: Elastic Moduli from Shape and Wrinkle Analysis. *Langmuir*, **29**, 12463–12471. doi:10.1021/la402322g (2013)

188. PESKIN, C. S. The immersed boundary method. *Acta Numerica*, **11**, 479–517. doi:10.1017/S0962492902000077 (Jan. 2002)

189. ALDER, B. J. & WAINWRIGHT, T. E. Molecular Motions. *Scientific American*, **201**, 113–126. doi:10.1038/scientificamerican1059-113 (1959)

190. ALDER, B. J. & WAINWRIGHT, T. E. Phase Transition for a Hard Sphere System. *The Journal of Chemical Physics*, **27**, 1208–1209. doi:10.1063/1.1743957 (1957)

191. ALDER, B. J. & WAINWRIGHT, T. Studies in molecular dynamics. I. General method. *The Journal of Chemical Physics*, **31**, 459–466. doi:10.1063/1.1730376 (1959)

192. ALLEN, M. P. & TILDESLEY, D. J. *Computer simulation of liquids* (Oxford university press, 1989)

193. FRENKEL, D. & SMIT, B. *Understanding molecular simulation: from algorithms to applications* (Academic press, 2001)

194. KADAU, K., GERMANN, T. C. & LOMDAHL, P. S. Molecular dynamics comes of age: 320 billion atom simulation on BlueGene/L. *International Journal of Modern Physics C*, **17**, 1755–1761. doi:10.1142/S0129183106010182 (2006)

195. ZHAO, G., PERILLA, J. R., YUFENYUY, E. L., MENG, X., CHEN, B., NING, J., AHN, J., GRONENBORN, A. M., SCHULTEN, K., AIKEN, C., *et al.* Mature HIV-1 capsid structure by cryo-electron microscopy and all-atom molecular dynamics. *Nature*, **497**, 643–646. doi:10.1038/nature12162 (2013)

196. BRINI, E., ALGAER, E. A., GANGULY, P., LI, C., RODRIGUEZ-ROPERO, F. & VAN DER VEGT, N. F. A. Systematic coarse-graining methods for soft matter simulations - a review. *Soft Matter*, **9**, 2108–2119. doi:10.1039/C2SM27201F (7 2013)

197. MÜLLER, E. A. & JACKSON, G. Force-Field Parameters from the SAFT-$\gamma$ Equation of State for Use in Coarse-Grained Molecular Simulations. *Annual Review of Chemical and Biomolecular Engineering*, **5**, 405–427. doi:10.1146/annurev-chembioeng-061312-103314 (2014)

198. MIE, G. Zur kinetischen Theorie der einatomigen Körper. *Annalen der Physik*, **316**, 657–697. doi:10.1002/andp.19033160802 (1903)

199. RAMRATTAN, N., AVENDAÑO, C., MÜLLER, E. & GALINDO, A. A corresponding-states framework for the description of the Mie family of intermolecular potentials. *Molecular Physics*, 1–16. doi:10.1080/00268976.2015.1025112 (2015)

200. LOBANOVA, O., AVENDAÑO, C., LAFITTE, T., MÜLLER, E. A. & JACKSON, G. SAFT-$\gamma$ force field for the simulation of molecular fluids: 4. A single-site coarse-grained model of water applicable over a wide temperature range. *Molecular Physics*, **113**, 1228–1249. doi:10.1080/00268976.2015.1004804 (2015)

201. AVENDAÑO, C., LAFITTE, T., GALINDO, A., ADJIMAN, C. S., JACKSON, G. & MÜLLER, E. A. SAFT-$\gamma$ force field for the simulation of molecular fluids. 1. A single-site coarse grained model of carbon dioxide. *The Journal of Physical Chemistry B*, **115**, 11154–11169. doi:10.1021/jp204908d (2011)

202. AVENDAÑO, C., LAFITTE, T., ADJIMAN, C. S., GALINDO, A., MÜLLER, E. A. & JACKSON, G. SAFT-$\gamma$ force field for the simulation of molecular fluids: 2. Coarse-grained models of greenhouse gases, refrigerants, and long alkanes. *The Journal of Physical Chemistry B*, **117**, 2717–2733. doi:10.1021/jp306442b (2013)

203. LAFITTE, T., AVENDAÑO, C., PAPAIOANNOU, V., GALINDO, A., ADJIMAN, C. S., JACKSON, G. & MÜLLER, E. A. SAFT-$\gamma$ force field for the simulation of molecular fluids: 3. Coarse-grained models of benzene and hetero-group models of n-decylbenzene. *Molecular Physics*, **110**, 1189–1203. doi:10.1080/00268976.2012.662303 (2012)

204. MÜLLER, E. A. & MEJÍA, A. Resolving Discrepancies in the Measurements of the Interfacial Tension for the CO2 + H2O Mixture by Computer Simulation. *The Journal of Physical Chemistry Letters*, **5**, 1267–1271. doi:10.1021/jz500417w (2014)

205. HERDES, C., FORTE, E., JACKSON, G. & MÜLLER, E. Predicting the adsorption of n-perfluorohexane in BAM P109 standard activated carbon by molecular simulation using SAFT-$\gamma$ Mie coarse-grained force fields. *Adsorption Science and Technology*, **33**. doi:10.1177/0263617415619528 (2015)

206. HERDES, C., SANTISO, E., JAMES, C., EASTOE, J. & MÜLLER, E. Modelling the Interfacial Behaviour of Dilute Light-Switching Surfactant Solutions. *Journal of Colloid and Interface Science*, **445**, 16–23. doi:10.1016/j.jcis.2014.12.040 (2015)

207. LOBANOVA, O., MEJIA, A., JACKSON, G. & MUELLER, E. SAFT-$\gamma$ force field for the simulation of molecular fluids 6: Binary and ternary mixtures comprising water, carbon dioxide, and n-alkanes. *Journal of Chemical Thermodynamics*, **93**, 320–336. doi:10.1016/j.jct.2015.10.011 (2016)

208. THEODORAKIS, P. E., MÜLLER, E. A., CRASTER, R. V. & MATAR, O. K. Superspreading: Mechanisms and Molecular Design. *Langmuir*, **31**, 2304–2309. doi:10.1039/C5SM02090E (2015)

209. MEJÍA, A., HERDES, C. & MÜLLER, E. A. Force fields for coarse-grained molecular simulations from a corresponding states correlation. *Industrial & Engineering Chemistry Research*, **53**, 4131–4141. doi:10.1021/ie404247e (2014)

210. VAN DER WAALS, J. D. *Over de Continuiteit van den Gas- en Vloeistoftoestand*. PhD thesis (University of Leiden, 1873)

211. KIRKWOOD, J. G. & BUFF, F. P. The Statistical Mechanical Theory of Surface Tension. *The Journal of Chemical Physics*, **17**, 338–343. doi:10.1063/1.1747248 (1949)

212. DEN OTTER, W. K. & BRIELS, W. J. The bending rigidity of an amphiphilic bilayer from equilibrium and nonequilibrium molecular dynamics. *The Journal of Chemical Physics*, **118**, 4712–4720. doi:10.1063/1.1543941 (2003)

213. BOEK, E. S., PADDING, J. T., DEN OTTER, W. K. & BRIELS, W. J. Mechanical Properties of Surfactant Bilayer Membranes from Atomistic and Coarse-Grained Molecular Dynamics Simulations. *The Journal of Physical Chemistry B*, **109**, 19851–19858. doi:10.1021/jp054372b (2005)

214. LORD KELVIN. Kinetic theory of the dissipation of energy. *Nature*, **9**, 441–444. doi:10.1038/009441c0 (1874)

215. BATTIMELLI, G. & FRENKEL, D. *Oral history interview with Berni J. Alder* http://www.aip.org/history-programs/niels-bohr-library/oral-histories/30662. 1990

216. PATRA, P. K., SPROTT, J. C., HOOVER, W. G. & HOOVER, C. G. Deterministic time-reversible thermostats: chaos, ergodicity, and the zeroth law of thermodynamics. *Molecular Physics*, **113**, 2863–2872. doi:10.1080/00268976.2015.1026856 (2015)

217. HOOVER, W. G., SPROTT, J. C. & PATRA, P. K. Ergodic time-reversible chaos for Gibbs' canonical oscillator. *Physics Letters A*, **379**, 2935–2940. doi:10.1016/j.physleta.2015.08.034 (2015)

218. HOOVER, W. G. An Appreciation of Berni Julian Alder. arXiv: 1510.05897 (2015)

219. STROGATZ, S. H. *Nonlinear dynamics and chaos* (Westview Press, 2014)

220. BUCKHEIT, J. B. & DONOHO, D. L. *Wavelab and reproducible research* (Springer, 1995)

221. CLAERBOUT, J. *Reproducible computational research: A history of hurdles, mostly overcome* http://sepwww.stanford.edu/sep/jon/reproducible.html

222. TORVALDS, L. & HAMANO, J. *Git: Fast version control system* http://git-scm.com. 2005

223. YAWS, C. L. *Thermophysical properties of chemicals and hydrocarbons* (William Andrew, 2008)

224. MCKINNEY, W. *Data Structures for Statistical Computing in Python* in *Proceedings of the 9th Python in Science Conference* (eds VAN DER WALT, S. & MILLMAN, J.) (2010), 51–56

225. MCKINNEY, W. *Python Data Analysis Library pandas: Python Data Analysis Library* http://pandas.pydata.org/. 2015

226. RONACHER, A. *Flask (A Python Microframework)* http://flask.pocoo.org/. 2015

227. JONES, R. *PEP 0301 – Package Index and Metadata for Distutilsckage Index and Metadata for Distutils* tech. rep. (Python Distutils-SIG, 2002)

228. ANDERSON, J. A., LORENZ, C. D. & TRAVESSET, A. General purpose molecular dynamics simulations fully implemented on graphics processing units. *Journal of Computational Physics*, **227**, 5342–5359. doi:10.1016/j.jcp.2008.01.047 (2008)

229. GLASER, J., NGUYEN, T. D., ANDERSON, J. A., LUI, P., SPIGA, F., MILLAN, J. A., MORSE, D. C. & GLOTZER, S. C. Strong scaling of general-purpose molecular dynamics simulations on GPUs. *Computer Physics Communications*, **192**, 97–107. doi:10.1016/j.cpc.2015.02.028 (2015)

230. THE TECPLOT TEAM. *Tecplot 360 EX 2016 User's Manual* Tecplot, Inc. (Bellevue, Washington, 2016)

231. CHILDS, H., BRUGGER, E. S., BONNELL, K. S., MEREDITH, J. S., MILLER, M., WHITLOCK, B. J. & MAX, N. *A Contract-Based System for Large Data Visualization* in *Proceedings of IEEE Visualization 2005* (2005), 190–198

232. BLENDER COMMUNITY. *Blender – a 3D modelling and rendering package* Blender Foundation (Blender Institute, Amsterdam, 2015)

233. R CORE TEAM. *R: A Language and Environment for Statistical Computing* R Foundation for Statistical Computing (Vienna, Austria, 2013)

234. WICKHAM, H. *ggplot2: elegant graphics for data analysis* (Springer New York, 2009)

235. KNUTH, D. E. Literate Programming. *The Computer Journal*, **27**, 97–111. doi:10.1093/comjnl/27.2.97 (1984)

236. TEIGEN, K. E., LERVÅG, K. Y. & MUNKEJORD, S. T. *Sharp interface simulations of surfactant-covered drops in electric fields* in *Fifth European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2010* (2010)

237. LERVÅG, K. Y., MÜLLER, B. & MUNKEJORD, S. T. Calculation of the interface curvature and normal vector with the level-set method. *Computers & Fluids*, **84**, 218–230. doi:10.1016/j.compfluid.2013.06.004 (2013)

238. BRUNSVOLD, A. L., ERVIK, Å. & ZHAO, H. *Experimental methods for investigating the discrete droplet impact phenomena of a model fluid relevant for LNG heat exchangers* in *ASME 2013 Fluids Engineering Division Summer Meeting* (2013), V01CT17A003–V01CT17A003

239. ERVIK, Å., HELLESØ, S. M., MUNKEJORD, S. T. & MÜLLER, B. *Experimental and computational studies of water drops falling through model oil with surfactant and subjected to an electric field* in *Proceedings of the IEEE 18th International Conference on Dielectric Liquids* (Bled, Slovenia, July 2014). doi:10.1109/ICDL.2014.6893172

240. GJENNESTAD, M. & MUNKEJORD, S. T. Modelling of heat transport in two-phase flow and of mass transfer between phases using the level-set method. *Energy Procedia*, **64**, 53–62. doi:10.1016/j.egypro.2015.01.008 (2015)

241. ERVIK, Å., PENNE, T. E., HELLESØ, S. M., MUNKEJORD, S. T. & MÜLLER, B. Influence of surfactants on the electrohydrodynamic stretching of water drops in oil. *Submitted to Physical Review Fluids*, (2016)

242. ERVIK, Å., LYSGAARD, M. O., HERDES, C., JIMÉNEZ-SERRATOS, G., MÜLLER, E. A., MUNKEJORD, S. & MÜLLER, B. A multiscale method for simulating fluid interfaces contaminated by large molecules such as asphaltenes. *Submitted to Journal of Computational Physics*, (2016)

243. ERVIK, Å. *Domain decomposition without the agonizing pain* http://dx.doi.org/10.6084/m9.figshare.1368581.v1. 2015

244. ERVIK, Å. *PETSc DMDA tutorial ex13* http://www.mcs.anl.gov/petsc/petsc-current/src/dm/examples/tutorials/ex13f90.F90.html. 2015

245. MCCALPIN, J. D. *STREAM web site* http://www.cs.virginia.edu/stream/. 2014

246. COLELLA, P. *Defining software requirements for scientific computing.* Presentation on the DARPA High Productivity Computing Systems program. 2004

247. LERVÅG, K. Y. & ERVIK, Å. in *Numerical Mathematics and Advanced Applications 2011* 209–217 (Springer, 2013)

248. SALAC, D. & LU, W. A Local Semi-Implicit Level-Set Method for Interface Motion. *Journal of Scientific Computing*, **35**, 330–349. doi:10.1007/s10915-008-9188-6 (2 2008)

249. MACKLIN, P. & LOWENGRUB, J. An improved geometry-aware curvature discretization for level set methods: Application to tumor growth. *Journal of Computational Physics*, **215**, 392–401. doi:10.1016/j.jcp.2005.11.016 (2006)

250. ERVIK, Å. *The local level-set extraction method for robust calculation of geometric quantities in the level-set method.* MA thesis (Department of Physics, Norwegian University of Science and Technology, 2012)

251. ERVIK, Å., MUNKEJORD, S. T. & MÜLLER, B. *Extending a serial 3D two-phase CFD code to parallel execution over MPI by using the PETSc library for domain decomposition* in *Proceedings of the 10th International Conference on Computational Fluid Dynamics In the Oil & Gas, Metallurgical and Process Industries* (Trondheim, Norway, June 2014). arXiv: 1405.3805

252. SANTINI, E., LIGGIERI, L., SACCA, L., CLAUSSE, D. & RAVERA, F. Interfacial rheology of Span 80 adsorbed layers at paraffin oil–water interface and correlation with the corresponding emulsion properties. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, **309**, 270–279. doi:10.1016/j.colsurfa.2006.11.041 (2007)

253. PELTONEN, L. J. & YLIRUUSI, J. Surface Pressure, Hysteresis, Interfacial Tension, and CMC of Four Sorbitan Monoesters at Water–Air, Water–Hexane, and Hexane–Air Interfaces. *Journal of Colloid and Interface Science*, **227**, 1–6. doi:10.1006/jcis.2000.6810 (2000)

254. OGATA, K. *Modern Control Engineering* 5th ed., 912 (Prentice Hall, New Jersey, 2009)

255. ERVIK, Å. & BJØRKLUND, E. The admissible surfactant distributions and velocities for small falling drops. *Submitted to Journal of Fluid Mechanics*, (2016)

256. CHANG, L. S. & BERG, J. C. The Effect of Interfacial tension Gradients on the Flow Structure of Single Drops or Bubbles Translating in an Electric Field. *AIChE Journal*, **31**, 551–557. doi:10.1002/aic.690310404 (1985)

257. LANGMUIR, I. The constitution and fundamental properties of solids and liquids. II. Liquids. *Journal of the American Chemical Society*, **39**, 1848–1906. doi:10.1021/ja02254a006 (1917)

258. GRIFFITH, R. The effect of surfactants on the terminal velocity of drops and bubbles. *Chemical Engineering Science*, **17**, 1057–1070. doi:10.1016/0009-2509(62)80084-0 (1962)

259. VERRUTO, V. J. & KILPATRICK, P. K. Water-in-model oil emulsions studied by small-angle neutron scattering: interfacial film thickness and composition. *Langmuir*, **24**, 12807–12822. doi:10.1021/la802095m (2008)

260. ERVIK, Å., SERRATOS, M. G. J. & MÜLLER, E. A. raaSAFT: a framework enabling coarse-grained molecular dynamics simulations based on the SAFT-$\gamma$ Mie force field. *Submitted to Computer Physics Communications*, (2016)

261. COWIE, J. & MCEWEN, I. Polymer-cosolvent systems: 5. Upper and lower critical solution temperatures of polystyrene in n-alkanes. *Polymer*, **24**, 1445–1448. doi:10.1016/0032-3861(83)90228-8 (1983)

262. ERVIK, Å., MEJÍA, A. & MÜLLER, E. A. Bottled SAFT: a web app providing SAFT-$\gamma$ Mie force field parameters for thousands of molecular fluids. *Submitted to Journal of Chemical Information and Modelling*, (2016)

263. LEE, S. & KIM, H. Liquid-liquid equilibria for the ternary systems sulfolane+ octane+ benzene, sulfolane+ octane+ toluene, and sulfolane+ octane+ p-xylene at elevated temperatures. *Journal of Chemical & Engineering Data*, **43**, 358–361. doi:10.1021/je9702246 (1998)

264. ERVIK, Å., LYSGAARD, M. O., HERDES, C., MÜLLER, E. A., MUNKEJORD, S. T. & MÜLLER, B. *A Multiscale Coarse-Grained Molecular and Continuum Model for the Flow of Droplets in Oil With Asphaltenes* Presented at the 2015 PetroPhase conference. 2015

265. LALANNE, B., VILLEGAS, L. R., TANGUY, S. & RISSO, F. On the computation of viscous terms for incompressible two-phase flows with Level Set/Ghost Fluid Method. *Journal of Computational Physics*, **301**, 289–307. doi:10.1016/j.jcp.2015.08.036 (2015)

266. PATIL, D. V., PREMNATH, K. N. & BANERJEE, S. Multigrid lattice Boltzmann method for accelerated solution of elliptic equations. *Journal of Computational Physics*, **265**, 172–194. doi:10.1016/j.jcp.2014.01.049 (2014)

267. DELLA ROCCA, G. & BLANQUART, G. Level set reinitialization at a contact line. *Journal of Computational Physics*, **265**, 34–49. doi:10.1016/j.jcp.2014.01.040 (2014)

268. HAN, S.-I., STAPF, S. & BLÜMICH, B. NMR imaging of falling water drops. *Physical Review Letters*, **87**, 144501. doi:10.1103/PhysRevLett.87.144501 (2001)

269. JACOBS, W. M., REINHARDT, A. & FRENKEL, D. Communication: Theoretical prediction of free-energy landscapes for complex self-assembly. *The Journal of Chemical Physics*, **142**, 021101. doi:10.1063/1.4905670 (2015)

# Appendix

# Paper 1

Journal of Computational Physics 257 (2014) 259–277

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

# A robust method for calculating interface curvature and normal vectors using an extracted local level set

Åsmund Ervik [a,b,*], Karl Yngve Lervåg [b], Svend Tollak Munkejord [a]

[a] SINTEF Energy Research, P.O. Box 4761 Sluppen, NO-7465 Trondheim, Norway
[b] NTNU, Department of Energy and Process Engineering, Kolbjørn Hejes v 1B, NO-7491 Trondheim, Norway

ARTICLE INFO

*Article history:*
Received 15 March 2013
Received in revised form 27 August 2013
Accepted 30 September 2013
Available online 10 October 2013

*Keywords:*
Level-set method
Curvature
Normal vector
Droplet-film interaction

ABSTRACT

The level-set method is a popular interface tracking method in two-phase flow simulations. An often-cited reason for using it is that the method naturally handles topological changes in the interface, e.g. merging drops, due to the implicit formulation. It is also said that the interface curvature and normal vectors are easily calculated. This last point is not, however, the case in the moments during a topological change, as several authors have already pointed out. Various methods have been employed to circumvent the problem. In this paper, we present a new such method which retains the implicit level-set representation of the surface and handles general interface configurations. It is demonstrated that the method extends easily to 3D. The method is validated on static interface configurations, and then applied to two-phase flow simulations where the method outperforms the standard method and the results agree well with experiments.



## 1. Introduction

Investigations of droplet collision phenomena have a long tradition in the study of multiphase flow, dating back to Lord Rayleigh [1] who in 1878 noted that raindrops can bounce off each other, and to Worthington [2] who in 1876 studied among other things the central jet that now bears his name. The early work predates the rise of computational studies, and consists of experimental studies that enabled a separation of the flow patterns into various regimes characterized by e.g. the Weber number and Ohnesorge number. A case which has long been the focus of study is that of a single droplet of one liquid, immersed in some other gas or liquid, and which collides with a deep pool of the first liquid. This could be e.g. a raindrop falling onto a pond, or a droplet of liquefied natural gas (LNG) merging with a pool of LNG in a liquefaction heat exchanger, so the case is interesting also from an industry point of view. Such a system may seem simple at first, but experimental and numerical studies have shown that varied phenomena such as coalescence, bouncing, jetting and partial merging occur. The system is also not fully understood yet; as an example, Thoroddsen et al. [3] have recently shown that for high impact velocities a turbulent boundary layer forms between the droplet and the pool after they merge.

In order to study such a case using computer simulations, it is necessary to use a precise interface tracking method to capture the physics before, during and after the collision. The level-set method (LSM) is a popular choice for interface tracking in studies of collisions, since its implicit formulation means that the method can handle the topological change which occurs when two interfaces merge. The LSM is very general, and apart from fluid dynamics it has been used for modeling such diverse phenomena as tumor growth [4], wildland fire propagation [5] and computer RAM production [6]. For a good introduction to the LSM, see e.g. [7]. The LSM originated from the seminal article by Osher and Sethian [8].

* Corresponding author at: NTNU, Department of Energy and Process Engineering, Kolbjørn Hejes v 1B, NO-7491 Trondheim, Norway.
*E-mail addresses:* asmund.ervik@sintef.no (Å. Ervik), karl.y.lervag@ntnu.no (K.Y. Lervåg), svend.t.munkejord@sintef.no (S.T. Munkejord).

0021-9991/$ – see front matter 
http://dx.doi.org/10.1016/j.jcp.2013.09.053

In two-phase flow simulations using the LSM, accurate interface curvature and normal vector information is vital in order to get good results. Standard methods exist for calculating these geometric quantities, but they fail when the interface topology changes, e.g. when two drops collide and merge. Several approaches have been used to remedy this flaw. The first approach to this problem is described by Smereka in [9]. He describes the problem briefly, and increases the numerical smoothing in the curvature discretization to lessen the effect. This is not an optimal solution, and Smereka notes on one of the simulations with merging interfaces that "most of the area loss occurs at the topology change". Several non-smearing approaches have subsequently been developed, by Macklin and Lowengrub [4,10], by Salac and Lu [11] and by Lervåg [12,13]. The methods by Macklin and Lowengrub and by Lervåg use curve fitting to obtain an accurate representation of the interface, while the method by Salac and Lu extracts several level-set functions each representing only a single body, and uses these to calculate the curvature.

The present work proposes a new method, which is an extension of previous methods, for calculating the curvature and normal vectors. The proposed method is based on the method by Salac and Lu, but it handles more general interface configurations and topological changes, as it considers only the local area around a point. The quality function introduced by Macklin and Lowengrub is used to restrict the use of the proposed method to those areas where it is needed, thus reducing the computational cost. As the proposed method uses no curve fitting, it extends easily to three dimensions, as demonstrated here. The proposed method is compared to the standard method for demanding cases where the analytical curvature is known; for such a case the proposed method gives errors of 1–2% where the standard method gives errors of $\mathcal{O}(1/\Delta x) > 100\%$. The proposed method is based on the work of Ervik [14].

The outline of this work is as follows: In Section 2, the theory of two-phase incompressible flow, the LSM and numerical methods are briefly reviewed. In Section 3, the proposed method is presented in detail. In Section 4, the method is validated on geometric test cases, and the results are compared to other methods. In Section 5, the results of two-phase flow simulations using the current method are reported and compared to experimental results. Finally, in Section 6, some concluding remarks are offered.

## 2. The level-set method and two-phase flow

The LSM is one of the more successful interface-capturing methods used in computational physics. Since its introduction by Osher and Sethian in [8], it has been used for numerous physical applications, as well as in computer graphics. Perhaps the main virtue of the LSM is how intuitive it is; in 2D it can easily be explained to anyone with a basic knowledge of multivariate calculus. This simplicity stems from the implicitness of the LSM, making the numerical implementation of the LSM relatively easy. The implicit formulation also means that changes in the interface topology are handled naturally. When comparing the LSM to other interface-tracking methods, such as the front-tracking method [15] where the interface is represented by piecewise continuous functions, the simplicity becomes especially clear.

The main disadvantage of the LSM, on the other hand, is that it is not a conservative method. During the course of a simulation, a fraction of fluid 1 may be converted to fluid 2 in an unphysical fashion. Various methods have been invented to circumvent this, e.g. the HCR-2 reinitialization method [16], so it is only a small effect presently. Interface-tracking methods may be conservative; an example of this is the volume-of-fluid (VOF) method, but then they typically have other disadvantages. In the VOF method, for instance, the advection equation cannot easily be solved, necessitating the use of interface-reconstruction methods [17]. Recent efforts have attempted to join the LSM and VOF in order to get the benefits of both methods; this approach seems to be fairly successful [18]. In a similar spirit, recent hybrid level-set/front-tracking methods have been developed [19] that retain the implicit definition of the interface while utilizing the front-tracking method to improve mass conservation and to compute the surface-tension forces in an accurate and robust manner.

We give here the formal definition of the level-set function used in the LSM. Let $\Gamma$ be the interface between two fluids, e.g. air and water, and $S$ be the computational domain where the fluids are confined. To represent this interface, we define a *level-set function* $\phi : S \to \mathbb{R}$ with the property

$$\Gamma = \{\mathbf{x} \mid \phi(\mathbf{x}) = 0\}. \tag{1}$$

This only defines the value of $\phi$ at the interface $\Gamma$, and not elsewhere. The common choice here is a signed distance function. Thus $\phi$ is fully specified by

$$\phi(\mathbf{x}) = \begin{cases} -\operatorname{dist}(\mathbf{x}, \Gamma) & \text{if } \mathbf{x} \text{ is inside } \Gamma, \\ \operatorname{dist}(\mathbf{x}, \Gamma) & \text{if } \mathbf{x} \text{ is outside } \Gamma. \end{cases} \tag{2}$$

Here, the function $\operatorname{dist}(\mathbf{x}, \Gamma)$ is the shortest distance from the point $\mathbf{x} \in S$ to the interface $\Gamma$. With this definition of the level-set function, the normal vector to the interface is given by

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}. \tag{3}$$

From this, the curvature is calculated by the well-known formula

$$\kappa = \nabla \cdot \mathbf{n} = \nabla \cdot \left( \frac{\nabla\phi}{|\nabla\phi|} \right). \tag{4}$$

With suitable discretizations of the derivatives involved, these quantities are easy to calculate numerically. This is often quoted as one of the nice features of the LSM, along with e.g. the very natural way the method handles topological changes [20]. In 2D, the standard discretization of the curvature is (see e.g. [21])

$$\kappa = \frac{\phi_{xx} + \phi_{yy}}{(\phi_x^2 + \phi_y^2 + \epsilon)^{1/2}} - \frac{\phi_x^2\phi_{xx} + \phi_y^2\phi_{yy} + 2\phi_x\phi_y\phi_{xy}}{(\phi_x^2 + \phi_y^2 + \epsilon)^{3/2}}. \tag{5}$$

Here, e.g. $\phi_x$ denotes the first derivative of $\phi$ in $x$-direction, calculated using standard central differences. However, when curvature and normal vector calculations are done during a change in the interface topology, this approach fails; the error in curvature is of the order $\mathcal{O}(1/\Delta x)$ [4]. In [9], Smereka notes that "One of the major advantages of level-set methods is their ability to easily handle topological changes. However for this problem we have found this not to be the case." It is this that the present method attempts to solve.

From the defining Eq. (2), $\phi$ is initialized at the start of a simulation. For a given velocity field $\mathbf{u}$, $\phi$ should be transported so that the interface follows the flow. This is done by solving the advection equation,

$$\frac{\partial \phi}{\partial t} = v|\nabla\phi| = -\mathbf{u}\cdot\nabla\phi. \tag{6}$$

Here $v$ is the velocity normal to the interface, and $\mathbf{u}$ is an *extrapolated* velocity field constructed using the method in [22]. This equation is not justified here, see e.g. [23].

Solving this equation will result in transportation of the interface, but it will also degrade the accuracy of the interface representation, as $\phi$ is deformed from a signed distance function. To avoid this, the level-set function is periodically *reinitialized*. We follow here the PDE-based approach introduced by Sussman, Smereka and Osher [23], which consists in solving

$$\frac{\partial \phi}{\partial \tau} + \mathrm{sgn}(\phi)\big(|\nabla\phi| - 1\big) = 0. \tag{7}$$

Here $\tau$ is a pseudo-time which is not related to the physical time in simulations. This approach is both computationally fast and accurate when used as here with a narrow-band approach. The extrapolation of the velocity field as used in Eq. (6) above is performed by solving a similar type of equation. These equations are solved using pseudo-CFL numbers of 1.0 for the velocity extrapolation and 0.5 for the reinitialization. It is noted that a numerical solution of the reinitialization equation needs accurate normal vectors at the interface.

A useful property of these equations is that the characteristics originate at the interface, meaning that solving the equations numerically for $N$ pseudo-time steps using a CFL-number of $C$ will yield a correct signed distance function $C \cdot N$ space steps away from the interface. This has led to the use of narrow-band methods, where the level-set function and other properties such as the curvature are only calculated and used in a narrow band around the interface. This reduces the computational time significantly.

In two-phase flow simulations, the LSM is coupled with the Navier–Stokes equations,

$$\nabla\cdot\mathbf{u} = 0, \tag{8}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{\nabla p}{\rho} + \nu\nabla^2\mathbf{u} + \mathbf{f}. \tag{9}$$

Here $\nu = \mu/\rho$ is the kinematic viscosity, while $\mu$ is the dynamic viscosity, $\rho$ is the density, $\mathbf{u}$ is the velocity field and $p$ is the pressure. $\mathbf{f}$ is any external force, such as gravity, and may be zero.

These equations hold for single-phase fluid flow, but can be extended to two-phase flow using different methods. In the present work, the ghost fluid method (GFM) [24] is used. This method prescribes jump conditions for e.g. the pressure across the interface based on the interface properties. The jump conditions used here are

$$[\mathbf{u}] = 0, \tag{10}$$

$$[p] = 2[\mu]\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n} + \sigma\kappa, \tag{11}$$

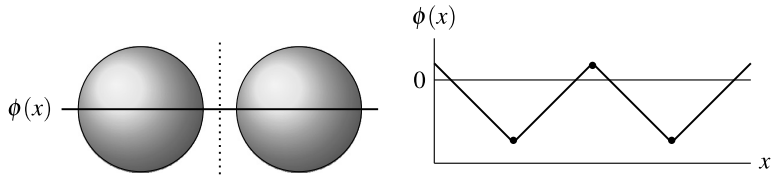
$$[\mu\nabla\mathbf{u}] = [\mu]\big((\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n})\mathbf{nn} + (\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{t})\mathbf{nt} \tag{12}$$

$$- (\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{t})\mathbf{tn} + (\mathbf{t}\cdot\nabla\mathbf{u}\cdot\mathbf{t})\mathbf{tt}\big), \tag{13}$$

$$[\nabla p] = 0, \tag{14}$$

based on [21]. Here, $\mathbf{t}$ is the tangent vector along the interface and $[\cdot]$ denotes the jump across an interface, that is $[\mu] \equiv \mu^+ - \mu^-$. Note that $\nabla\mathbf{u}$ and (e.g.) $\mathbf{nt}$ are rank-2 tensors. The pressure must also be decoupled from the velocity field in order to enable a numerical solution of the Navier–Stokes equations; we use here the projection method due to Chorin [25]. This gives a Poisson equation for the pressure which can be solved using freely available numerical libraries. The PETSc library is used here [26].

In the present numerical implementation, SSP-RK schemes [27,28] are used for the time integration, while the WENO method [29] is used for the spatial discretizations. To determine the time step dynamically, we use the CFL-criterion given by Kang et al. [21].




(a) Droplets in near contact (b) A slice of the level-set function

**Fig. 1.** (a) Two droplets in near contact. The dotted line marks a region where the derivative of the level-set function is not defined. (b) A one-dimensional slice of the level-set function. The dots mark points where the derivative of $\phi$ is not defined.

## 3. The local level-set extraction (LOLEX) method

### 3.1. Introduction

Calculating the curvature $\kappa$ of the interface between two phases is important, since it appears in the Young–Laplace formula for the capillary pressure, $\Delta p = \sigma\kappa$. Its value is used in e.g. the ghost fluid method (GFM) (Eq. (11)), or other methods of enforcing the jump conditions. The normal vectors to the interface are also important, e.g. when advecting the level-set function and when reinitializing it. Calculating these geometric quantities is straightforward in theory, using Eqs. (3) and (4) to compute them from the level-set function.

However, as is often the case, in practice it is not so straightforward. The problems arise when the distance between two interfaces is of the order $\Delta x$. This is illustrated in Fig. 1. The derivatives of $\phi$ are not defined at the kinks. As a result of this, the numerical stencils approximating the derivatives of $\phi$ will often produce large, erroneous values. When this happens, the curvatures and normal vectors will be erroneous. For the curvature, this error is of order $\mathcal{O}(1/\Delta x)$, which can be several orders of magnitude larger than the correct curvature value. It should be stressed that additional grid refinement does not solve this problem; e.g. for the simulation of colliding drops, one would have to continue refining the grid *ad infinitum*.

The earliest non-smearing approach to this problem, by Macklin and Lowengrub [4], uses a modification of the directional differences for points close to kinks, along with a mesh refinement for these points. The same authors introduced a curve-fitting method instead in [10], which is said to be an improvement on the directional differences and a simplification. The latter version will be referred to as the MLM (Macklin and Lowengrub method). Further improvements to this method, and adaptations to an on-grid framework (i.e. calculating the curvature at the grid points, not at the interface), have been developed by Lervåg [12,13]. These methods give good results in 2D, but are difficult to extend to 3D simulations due to the use of curve-fitting.

An alternative approach to the problem is due to Salac and Lu [11], and will be referred to as the Salac and Lu method (SLM). This approach extracts separate bodies represented by the level-set function into their own, separate distance functions. Only the negative parts of the level-set function are extracted, the positive parts are reconstructed through reinitialization. This procedure removes all kinks that are caused by two or more bodies that are close to each other. For a review and comparison of the SLM, MLM and the method by Lervåg, see Lervåg and Ervik [30]. It should also be noted that the recent article by Focke and Bothe [31] discusses a similar issue, in the context of thin lamellae which form when liquid drops collide off-center. The authors introduce a method which resembles the SLM, but which also has the ability to add small amounts of liquid to the lamella region, preventing a numerical rupture.

The method considered here is a further development of the SLM. It is referred to as the local level-set extraction method, or LOLEX method in short. The reason why the SLM is insufficient in some cases, as well as the details of the present method, is given below. Suffice it to say at this point that the present method is more general, so it applies both to the cases considered by Salac and Lu and those considered by Focke and Bothe (except the stabilization of thin lamellae which the latter introduce).

Another recently presented approach is due to Trontin et al. [32], who consider a hybrid particle/level-set method. Their approach is to use the information from the tracking particles to calculate the curvature and normal vectors, with good results. This can obviously not be applied to a pure level-set method as discussed here, or e.g. a coupled level-set/VOF (CLSVOF) method as has recently become popular [18].

The previously mentioned work by Shin et al. [19] which introduces a hybrid front-tracking/level-set method is another interesting approach. The ability of their method to conserve mass globally as well as locally is impressive, and the handling of thin filaments is better than the method proposed in this paper. As with the approach due to Trontin et al., this method cannot be applied to a pure level-set framework, and integrating it into an existing level-set based code would be arduous. In comparison, the method proposed in this paper can be implemented into a level-set framework with less than 500 lines of code.

An approach which has not been considered here, or by other authors in the context of level-set methods as far as we are aware, is the use of filtering. Vliet and Verbeek [33] study the estimation of curvature from a discretely sampled greyscale image, using derivative-of-Gaussian filters, and note that this outperforms a traditional curvature estimate analogous to Eq. (5).

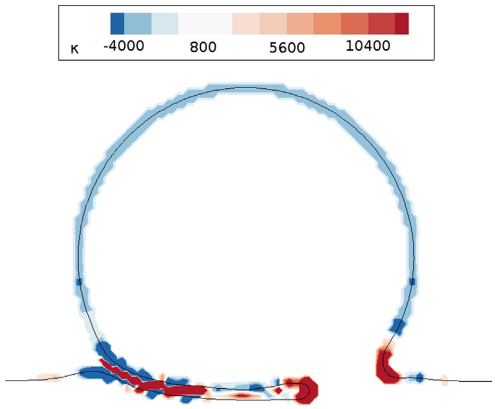



**Fig. 2.** The curvature field plotted for the SLM. Note the red curvature field inside the air finger between the drop and the pool, which is incorrect. The color should be light blue in this area. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The idea of Salac and Lu, on which the present method is based, is simple when compared to the curve-fitting scheme used by Macklin and Lowengrub [4] and later by Lervåg [12,13]. This simplicity is more in keeping with the "spirit" of the level-set method: the LSM is an implicit alternative to front-tracking methods that employ curve fitting, and this implicitness makes extending to higher dimensions straightforward. In the same fashion, the SLM is easily extensible to 3D, while the methods employing curve fitting are not. There are, however, some drawbacks to the Salac and Lu method as well.

The primary issue stems from the fact that the Salac and Lu method is aware of the global topology of the interface. A problematic area, with a kink in the level-set function close to $\phi = 0$, can be caused either by two bodies in close proximity or by a single body folding back onto itself. In the latter case, as illustrated in Fig. 2, the Salac and Lu method falls back to the standard discretization, and the calculated curvature will be erroneous. This may seem like an edge case not worth considering, but simulations have shown that this often happens, e.g. when a falling droplet merges into a pool. As pointed out by Smereka [9], errors like these can be the main contribution to unphysical area loss in a simulation. Another situation where this would often be the case is in tumor simulations like those performed by Macklin and Lowengrub, as can be seen in e.g. [4, Fig. 6].

### 3.2. The idea of the LOLEX method

The method presented here tries to combine the best of the SLM with the best of the MLM. As illustrated in the previous section, the SLM is aware of the global topology of the interface, which is problematic in some cases. The MLM does not have this problem, as its curve fitting considers only the local area, but as previously stated it does not extend easily to 3D. A natural workaround to the "global awareness" is to make the Salac and Lu method consider only the local topology; say, a $10 \times 10 \times 10$ cube around the point where we calculate the curvature.

In the following, we assume the level-set function to be located on a uniform mesh on a single CPU. The proposed method can be adapted in a straightforward manner both into a domain-decomposition and a mesh-refinement framework. We do not discuss this in further detail here.

Since the SLM relies on reinitialization to remove kinks, a potential problem with this approach is computational efficiency, as reinitialization can be time-consuming. To avoid this problem, we want to use the standard discretization as much as possible, only resorting to the LOLEX method when we have to, i.e. when kinks in $\phi$ are close to the interface. To easily identify kinks, we use the quality function $Q(\mathbf{x})$ which was introduced by Macklin and Lowengrub in [4]. It is defined as

$$Q(\mathbf{x}) = \left\| 1 - \nabla\phi(\mathbf{x}) \right\|_2, \tag{15}$$

i.e. the deviation of $\phi$ from a signed distance function, measured with the 2 norm. If $\max(Q(\mathbf{x}_{i,j,k})) > \eta$ for $\mathbf{x}_{i,j,k}$ in a $3 \times 3 \times 3$ cube around the current grid point, we use the LOLEX method. A value of $\eta = 0.005$ is used here, and is seen to perform well. That is, the number of grid points where the LOLEX method is used becomes small compared to the total number of grid points. This keeps the computational cost low. The effect of varying $\eta$ can be seen in Fig. 10, as discussed in Section 4.2

To further decrease the computational cost, we use the "narrow band" level-set method introduced in [34]. This means that quantities such as the curvature are only calculated in a narrow band around the zero level set, where they are needed.

Having briefly presented the idea behind the present method and the scope in which it will be used, we give here a step-by-step outline of it, see Fig. 3. 2D notation is used for clarity, but all steps are easily extensible to 3D. In this outline, a few arrays are introduced for storing data: `lookphi` is a copy of the global $\phi$ for the local area we are considering, `bodies` indicates the bodies present using increasing integers, and `locphi` holds the local $\phi$s that are extracted from the

↪ Loop over the computational domain using indices `i,j`.
↪ If ($\mathbf{x}_{i,j}$ not close to interface) do nothing. A point is defined as close to an interface if all $\phi(\mathbf{x}_{n,m})$ for $(n,m) \in [i-1, i+1] \times [j-1, j+1]$ is either negative or positive.
↪ Else if ( $Q(\mathbf{x}_{n,m}) \leq \eta \ \forall (n,m) \in [i-1, i+1] \times [j-1, j+1]$ ) use ordinary method.
↪ Else use LOLEX method:
  ↪ Copy $\phi$ in a `[-1,ilmax+2]*[-1,jlmax+2]` square around `i,j` into the `lookphi` array.
  ↪ Identify the bodies present in the `[0,ilmax+1]*[0,jlmax+1]` square, store this in the `bodies` array.
  ↪ For each body, extract the relevant part of the `lookphi` array into `locphi(:,:,bodyno)`. This array has 3 ghost cells on the boundary outside `ilmax*jlmax`; these are not used until the extrapolation further down. Extracting means:
    – copying `lookphi` for the internal points of *this* body
    – copying `lookphi` for external points that are not next to more than one body
    – explicitly reconstructing the signed distance for external points that are next to more than one body
    – setting a value of `2*dx` for all other points
  ↪ Once the `locphi` array has been filled for all bodies, the values are extrapolated into the ghost cells. The extrapolation is zeroth-order, as will be explained further down.
  ↪ The `locphi` array is then reinitialized for all bodies. This erases the problematic kink, as well as the value of `2*dx` which was set previously. Thus this value is unimportant, as long as it is $> 0$.
  ↪ Using these local $\phi$'s, the curvature and normal vectors can be calculated for each body. The curvature and normal vectors corresponding to the body which is closest to the current grid point are used.

**Fig. 3.** A step-by-step outline of the LOLEX algorithm.

global $\phi$ and then refined into more accurate representations of the local bodies present. The quantities `ilmax`, `jlmax` and `klmax` represent the number of grid points, in the $x$, $y$ and $z$ directions respectively, of the *local* grid. The values of `ilmax`, `jlmax`, `klmax` are all set to 7 in the simulations performed here. Their values are independent of the global grid size. Sensible values of these are between 5 and 11; since they must be odd, smaller than 5 gives too low resolution, and larger than 11 starts eliminating the advantage of using a localized method. The value of 7 used here gives good results, and increasing it to 9 gives only a small change while increasing the computational cost. In the limit `ilmax` $\rightarrow$ `imax` etc. the method of Salac and Lu is recovered.

The steps in the algorithm that warrant further comments are: identifying the bodies present, explicitly reconstructing the signed distance, extrapolating to the ghost cells, and reinitializing. These will be considered further in the next section and subsections.

### 3.3. Details of the method

Some steps of the algorithm outlined need further explanations. This is either because they are too technical to be fully described in the previous short outline, or because they have not been properly motivated yet. The steps that will be considered are identifying the bodies present (Section 3.3.1), explicitly reconstructing the signed distance (Section 3.3.2), extrapolating to the ghost cells (Section 3.3.3), and reinitializing (Section 3.3.4).

#### 3.3.1. Identifying the bodies present

To identify the bodies present, a recursive routine is used, which starts at a seed point in a body and iterates through the entire body, marking it as a body in the `bodies` array. The seed point is found by scanning the computational domain for points with $\phi < 0$. The recursive routine is called `bodyscan` here. The `bodies` array starts with a value of `unchecked`, and bodies found are marked using increasing integers. The recursive subroutine will have marked the entire first body when its first call returns.

A final point to note about the routine given here is that even though a recursive subroutine is used, memory usage will not be problematic. This is because the routine operates on a small array whose size is independent of the grid size. In 3 dimensions and with the presently used size of the local area, the array `bodyscan` would have `11*11*11` $= 1331$ elements. This routine can maximally be called 1331 times, giving a worst-case memory consumption of 13.5 MB. In reality this number would typically be less than half of that. This will not cause memory problems, although it is too large to fit in the CPU cache for some processors. The performance impact has not been tested here, as the 3D calculations are only considered as a proof-of-concept, and have not been optimized for speed. In 2D the memory use is naturally much smaller.

#### 3.3.2. Explicit reconstruction of the signed distance

For some points with $\phi > 0$, two or more bodies are within $\Delta x$ of the point. This means that the value of $\phi$ is probably incorrect, since it has to be the distance to two separate bodies at the same time. We will call such points "dependent points". These points are found using the `bodies` array: if this array has more than one unique positive integer value in the four points adjacent to the present point, it is dependent. Because $\phi$ is likely incorrect for dependent points, we discard its value, and instead explicitly reconstruct the distance to the relevant interface. The procedure used is due to Adalsteinsson et al. [35].

When we consider such a dependent point, it lies right next to two interfaces. When reconstructing the distance, only one interface is of interest, so the other one is momentarily removed. Note that the signed distance is always positive for exterior points, so it is just the normal distance.

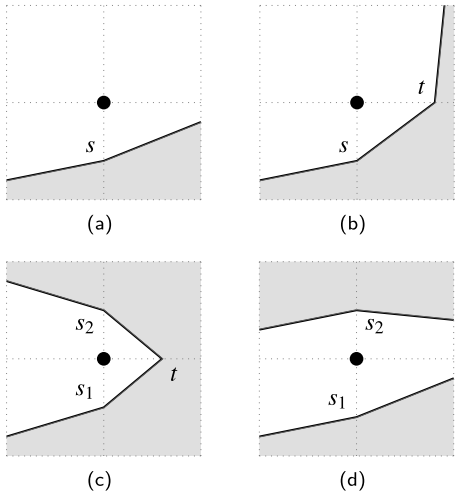



**Fig. 4.** Cases for the neighborhood of a point.

The procedure in [35] is as follows. The point $(i, j)$ which we are considering is next to the interface of current interest. We ignore all other interfaces. Up to rotational symmetry, there are four possible cases. An illustration of these cases can be seen in Fig. 4.

We examine the four cases (a–d) more closely:

a The interface crosses one of the lines from $(i, j)$ to its four neighbors. In this case, we use the distance to the interface along this line as our distance. This distance is given by

$$s = \Delta y + \phi(i, j-1) \tag{16}$$

where we have assumed that $(i, j-1)$ is the neighbor on the other side of the interface. Since this neighbor is an internal point, it has $\phi < 0$. The distance to the interface is the distance to the neighboring grid point $(\Delta y)$ minus the distance from that grid point to the interface, which gives this formula. It is best to use only the $\phi$-value inside the body, since it is less likely to be distorted.

b The interface crosses two of the lines, and these two lines make out a corner of the $2 \times 2$ grid around $(i, j)$. In this case we use the shortest distance to the straight line between the two points of intersection. The distance $d$ is given by the formula

$$\left(\frac{d}{s}\right)^2 + \left(\frac{d}{t}\right)^2 = 1. \tag{17}$$

As long as $s^2 + t^2 \neq 0$ this equation can be solved, and the positive solution is

$$d = \frac{st}{\sqrt{s^2 + t^2}}. \tag{18}$$

If we have $s^2 + t^2 = 0$, then $s = t = 0$, so it is obvious that the distance to the interface is $d = 0$.

c The interface crosses three lines. We construct the two straight lines between the points of intersection, and use the shortest distance to either of these two lines, given by

$$\left(\frac{d}{\min(s_1, s_2)}\right)^2 + \left(\frac{d}{t}\right)^2 = 1. \tag{19}$$

d The interface crosses two lines. These lines are on opposite sides of the point $(i, j)$. In this case, we use the shortest of the two distances, so $d = \min(s_1, s_2)$.

These formulae can be extended to three dimensions, where the possible cases are more numerous. In 3D, the central point has two additional neighbors. This means there are more variations in addition to the cases considered above. This is not considered in detail here.

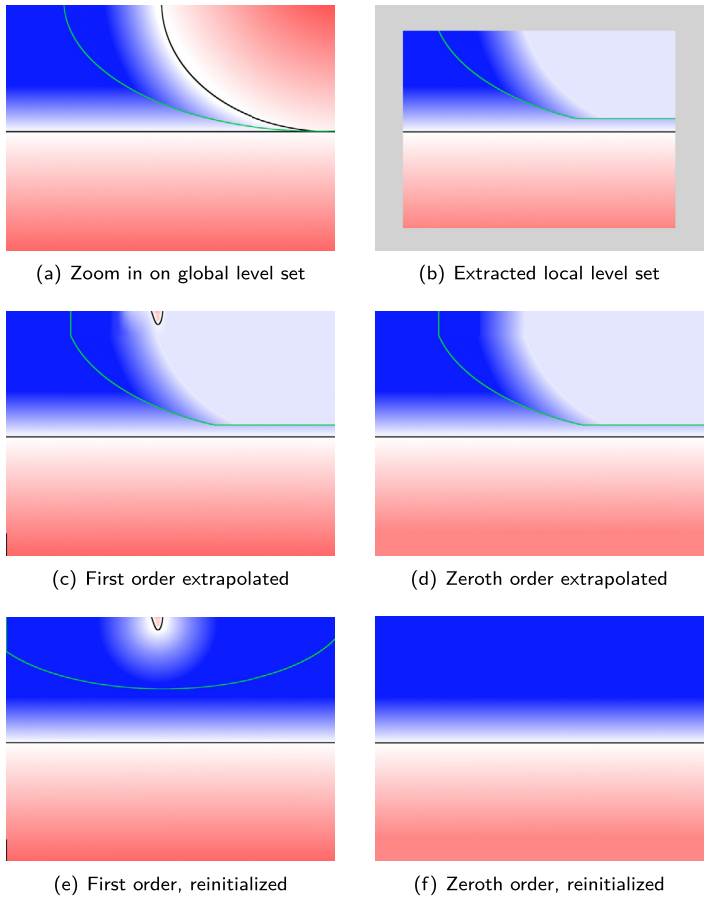



**Fig. 5.** Extraction, extrapolation and reinitialization of the local level set is shown, for the lower body in (a). Red indicates a negative value, blue a positive value, and white indicates zero. The green lines indicate kinks in the level set function, and the black lines are the zero level sets. A detailed explanation of the figures is given in Section 3.3.3. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

### 3.3.3. *Extrapolation*

After the interior of the `locphi` array has been filled, the ghost cells must be filled before we can reinitialize the local $\phi$. Two ways of doing this are illustrated in Fig. 5. A first approach is to use linear extrapolation, which should work well since $\phi$ is a linear function in 1D. However, it turns out that this does not work. A fundamental property of the reinitialization equation (7) is that its characteristics originate at the interface $\phi = 0$. This is why the present method (and the SLM) works – we only need a few cells directly next to the interface to have the correct value of $\phi$, and reinitialization will fix the rest. It also means that reinitialization will never move the position of the interface, which is a desirable property in general.

The problem with linear extrapolation occurs when we extrapolate starting on the opposite side of the kink from the interface. In this case, the values of the local $\phi$ are tending towards 0 from above, which means that extrapolation can reintroduce the other body (which we removed in the first place). When this happens, reinitialization cannot fix the values beyond the kink, since it cannot move the interface reintroduced by extrapolation. A straightforward alternative is to use a zeroth-order extrapolation. This means simply copying the values along the edges into the ghost cells. It is obvious that this will never cross $\phi = 0$, so reinitialization works as intended.

The difference between these two is shown in Fig. 5. In (a), a zoom in on the global level set of a droplet touching a pool is shown. In (b), the local level set of the lower body (the pool) is shown after extraction and explicit reconstruction. Here, the values on the edges are not set, indicated in grey. In (c), the same is shown after first-order extrapolation, and in (d) after zeroth-order extrapolation. In (e), the first-order extrapolated $\phi$ is shown reinitialized, and in (f) the zeroth-order extrapolated $\phi$ is shown reinitialized. Note in particular that in (e), a kink still exists after the entire procedure (green line), so the geometric quantities calculated would still be wrong if the derivatives cross the kink.

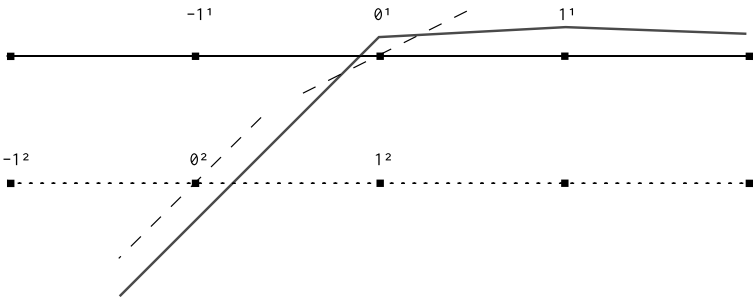



**Fig. 6.** Why we reinitialize from a lower level set: At the lower level set, indicated by the dotted line, values of e.g. $\nabla\phi$ are more accurate at the grid point which is closest to the grey line than for the zero level set. The grey line indicates the local level-set function $\phi$. The dashed lines indicate $\nabla\phi$ calculated using central differences.

The corner cells on the boundaries must also be set. Here, these all get the value from the corresponding corner of the internal grid.

*3.3.4. Reinitialization*

When the extracted local level-set has been extrapolated, it must then be reinitialized before the geometric quantities are calculated. This is essential in order to have good values of the level-set function outside the interface. The entire LOLEX method hinges on the fact that reinitialization restores the local level-set to a signed distance function, so that ordinary discretizations will not give errors. This is not entirely straightforward, however.

When reinitializing, we require at least some points on either side of the interface with decent $\phi$-values, i.e. $\phi$ being the signed distance to the interface. In addition to this, we need to know the smeared sign function, and most crucially, the normal vectors at the interface. Thus we are faced with a bootstrapping problem: accurate normal vectors are required in order to accurately calculate the normal vectors. This is only a problem when the global interfaces are very close; when there is a moderate distance (i.e. more than one grid point between the interfaces), the normal vectors can be calculated at the interface using the local level-set.

The solution to this conundrum is to exploit the redundant information which is stored in the level-set function. To illustrate this redundancy, imagine that you are walking along a normal vector to the interface. At each grid point you pass, you are told the current distance to the interface. As long as you do not pass any kinks, this information is redundant: using the value at the first grid point you pass, you can calculate the value at the next grid point, and the one after that, given that you know the grid spacing.

What this means for the present case is that we have information inside the current body that we can use. Most importantly, we can calculate the normal vectors without problem for internal points. This means that we can reinitialize a level set different from $\phi = 0$, e.g. $\phi = -0.8\Delta x$, and get essentially the correct $\phi$ afterwards. We are not guaranteed to get exactly the correct $\phi$, but as we cannot obtain the correct $\phi$ anyway, we will settle for a good approximation. An illustration of this in 1D is shown in Fig. 6, where the extracted local level-set function $\phi$ is shown in grey. Note that e.g. the value of $\nabla\phi$ at the grid point $0^2$, shown with a dashed line, is much closer to 1 than the value at the grid point $0^1$. When the lower level set is used, we momentarily move the interface further to the left in this figure, so the grid point $0^2$ is closest to the interface. It is obvious that we have a better chance of restoring a signed distance function with the correct location of the interface if we reinitialize from the lower level set.

The value of $-0.8\Delta x$ used here gives the most accurate results. If the value is too close to zero, the benefit of reinitializing from a lower level set is reduced. However, if the value is too large, we risk having this lower level set too close to the edges of the local domain, and we increase the potential error caused by reinitializing from a different level than zero. The optimal result is afforded by choosing a value somewhat below $-0.5\Delta x$, since this ensures that the grid point $0^2$ is always closest to the interface, while minimizing errors from the edges of the domain.

Another problem solved by this is the fact that the values directly outside the zero level set may be incorrect in some cases. In particular, this happens when an outside grid cell is not flagged as dependent, but its value of $\phi$ still deviates from that dictated by a signed distance function. Tests have shown that this sometimes occurs, and that it distorts the reconstructed local level set.

Reinitializing from a different level may sound somewhat complicated to do, but the implicit formulation springs to the rescue again. To reinitialize from a lower level set, we simply add a positive constant to $\phi$ at every local grid point, call the reinitialization routine on this $\phi$, and then subtract the same constant from the reinitialized $\phi$. The effect of this is illustrated in Fig. 7, which is an extreme case. Here, reinitialization of two very close bodies (concentric circles) has distorted the global level-set function close to and outside the interfaces. The reinitialized local level-set function is also wrong, but the one which is reinitialized from a lower level set gives a much smoother representation of the interface, which agrees with the contour lines further into the body. This smoother representation will, in turn, give a significantly more accurate curvature. A plot of the curvature calculated with and without this improvement is shown in Fig. 8 for the concentric circles

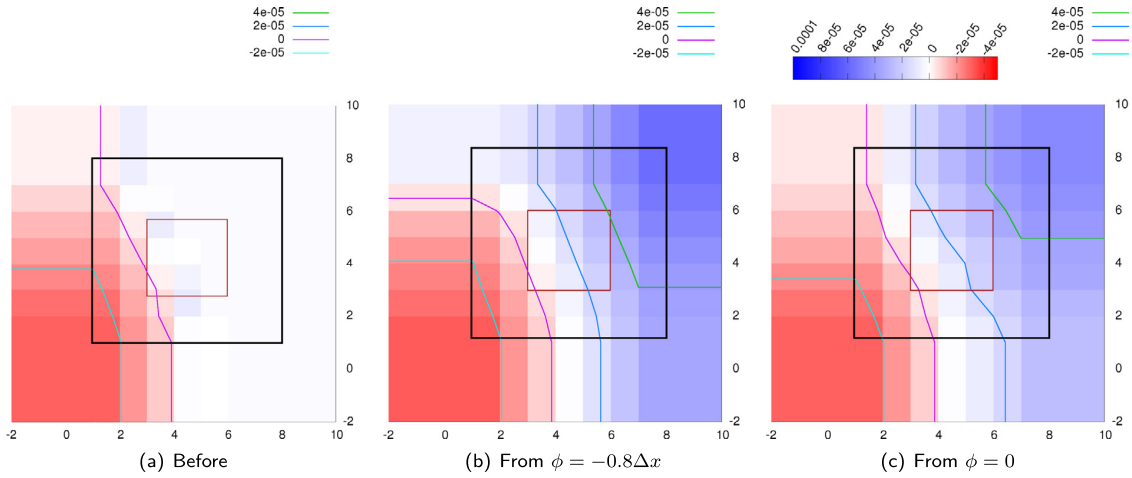



**Fig. 7.** The LOLEX method on a global level set which is distorted due to reinitialization of very close bodies. The global bodies are two concentric circles. (a) Local $\phi$ before reinitialization. (b) Local $\phi$ reinitialized from $\phi = -0.8\Delta x$. (c) Local $\phi$ reinitialized from $\phi = 0.0$. The black square indicates the boundary to the ghost cells, and the red square indicates the $3 \times 3$ central points that are used in the final curvature calculation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

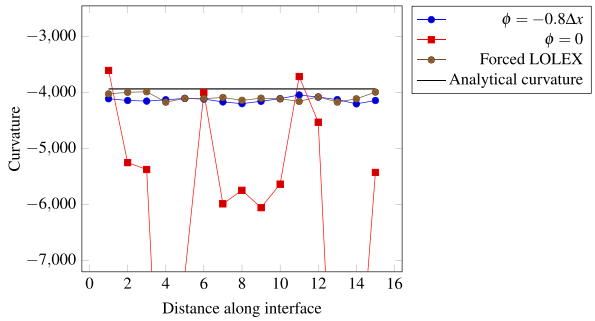



**Fig. 8.** Lineplots of the curvature along the interface when reinitializing from both the zero level set and a lower level set. Also shown are the curvature calculated when forcing use of the LOLEX method on a single interface, and the analytical curvature.

case; this global interface configuration can also be seen in Fig. 11 further down. This plot shows the curvature along the inner circle. It is seen that the improvement is large, particularly in this case when two interfaces are close. The curvature calculated using the standard method is not shown, as it is outside the $y$-axis range in this figure.

While the curvature calculated using the LOLEX method is close to the analytical value, there is still a more or less constant error of 1–2%. It turns out that this error is caused by the reinitialization of the local level set, as is indicated in this figure as well. The line captioned 'Forced LOLEX' shows the LOLEX method used on a single interface corresponding to the inner circle. Here, the level-set function is correct and the standard method gives an error for the single interface which is smaller than the line width in this figure. When we force the use of the LOLEX method, the only difference from the standard method is the extrapolation and reinitialization, meaning that these must be the culprits. To mitigate this, a more accurate reinitialization procedure could be used, e.g. the HCR method due to Hartmann et al. [16].

### 3.3.5. *Parameters of the method*

In the LOLEX method as presented here, there are a number of parameters that can be varied. An overview of these is given here, along with the values used presently, and sensible ranges, in Table 1.

After the local level sets have been extracted correctly, the standard discretizations can be used to calculate the normal vector and curvature. As the curvature and normal vector cannot be multiply defined at a single grid point, we must combine the information from different local level sets. To do this, we simply select the one corresponding to the interface which is closest to the central point.

As the present method uses reinitialization on a local grid for each grid point where it is used, the performance impact of the method could become large. To avoid this, the quality function $Q(\mathbf{x})$ is used to restrict the use of the method. In a typical falling drop simulation, the present method will only be used in a small percentage of the total number of time

**Table 1**
Parameters used in the LOLEX method, along with values used and sensible ranges.

| Parameter | Value | Sensible range |
|---|---|---|
| Local grid size | 7 | 5–11 |
| Gradient threshold $\eta$ | 0.005 | 0.01–0.001 |
| Reinit. level set | $-0.8\Delta x$ | $-1.0\Delta x$ to $-0.5\Delta x$ |

steps, and even then, it will typically not be used for all points along the interface. This means the computational cost of the present method has a low impact on the total runtime of a simulation. We note that the computational cost is lower than in the method of Salac and Lu, since that applies reinitialization to the entirety of the bodies present.

### 3.4. Summary

In this section the presently used LOLEX method has been described in detail. The method is used for grid points where the level-set function deviates from being a signed distance function, where it extracts one or more local level sets, removes any kinks in these by use of reinitialization, and finally uses these local level sets to calculate the curvature and normal vectors. The values corresponding to the interface which is closest to the current grid point is used.

The method is motivated in that it is more general than the previous method by Salac and Lu [11], handling bodies which fold back onto themselves, and it extends more easily to 3D than the previous methods by Macklin and Lowengrub [4,10] and by Lervåg [12,13], which use curve-fitting schemes. The parameters of the method are given in Table 1. Results, both for static and dynamic simulations, are given in the next sections.

## 4. Geometric results

In order to test the LOLEX method, some static interface configurations were used that replicate typical situations occurring in simulations of droplet collisions.

### 4.1. Circles and straight interfaces

The first test case consists of three circles and a straight-lined interface, where two of the circles and the straight-lined interface are joined together. The results for this case are shown in Fig. 9 for the LOLEX method, the SLM, and the standard method. In this figure, the interfaces are shown as black lines, and the color indicates the curvature. The background curvature of 0 is indicated in white, blue indicates a negative curvature and red indicates a positive curvature. The figure illustrates that the standard method produces positive unphysical curvatures several places, both between the circles and the straight interface and between circles. The Salac and Lu method remediates the situation somewhat, but still has problems where the circle folds back onto the straight interface, and at the bottom of the middle circle, which is particularly close to the straight interface. The LOLEX method produces positive curvatures only where they are expected and needed.

### 4.2. Droplet falling onto a pool

In order to compare the behavior of the LOLEX and the standard method for different interface separations, a test case was considered which mimics a droplet falling onto a pool. In this case, a 0.2 m diameter circle and a horizontal line were initialized in a 1 m $\times$ 1 m domain. The separation between the circle and the line was varied from $3.6\Delta x$ down to $0\Delta x$ in increments of $0.1\Delta x$. For each separation, the curvature was calculated at all points within the narrow band close to the interfaces, and the supremum-norm $\|\kappa\|_\infty$ of the curvature values was calculated. This was done using the standard and the LOLEX method, for grid resolutions of $64 \times 64$, $256 \times 256$ and $1024 \times 1024$. The analytical curvature is 10 for the circle and 0 for the line, so the supremum norm should be close to 10. The results are shown in Fig. 10.

As is seen in this figure, the standard method returns the value used in regularizing the curvature, $\|\kappa\|_\infty = \frac{1}{\Delta x}$, when the interface separation becomes smaller than about $2.4\Delta x$. Increasing the grid resolution does not improve the situation. Note that the $y$-axis in this plot is logarithmic. Meanwhile, the LOLEX method gives decent values close to the analytical value of 10 all the way up to when the interfaces merge, which happens at a separation of $0.2\Delta x$. It is seen that the small deviations for the LOLEX method are reduced when the grid resolution is increased.

A final thing which may be illustrated using this figure is the effect of the parameter $\eta$. This parameter indicates how much the level-set function $\phi$ has to deviate from being a signed distance function before we switch from using the standard method to the LOLEX method. In the circle-line case in Fig. 10 the value of $\eta = 0.005$ (used throughout this paper) triggers the switch for the first point when the distance is $4.2\Delta x$. Using $\eta = 0.01$ as in [10], the switch is made at $3.9\Delta x$. Both these distances are in the region where the standard method gives good answers, so the LOLEX method is not very sensitive to the precise value of $\eta$ as long as it is in this range.

In addition to the curvature, accurate normal vectors close to the interface are desirable in level-set simulations. The importance in reinitialization has been suggested above, coming from the fact that normal vectors are used in finding the

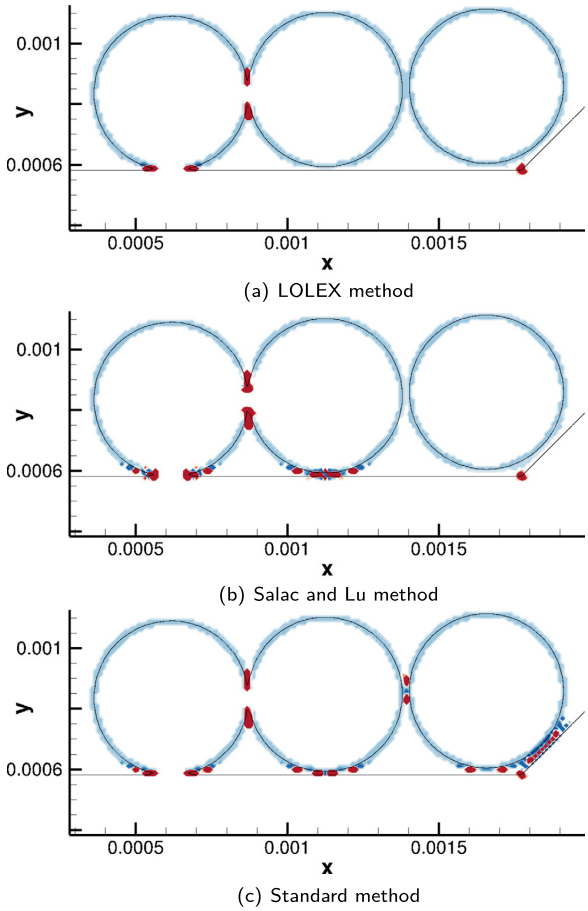



**Fig. 9.** Comparison of curvature calculation methods for circles and straight interfaces. The color indicates the curvature; white is zero, blue is negative and red is positive. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

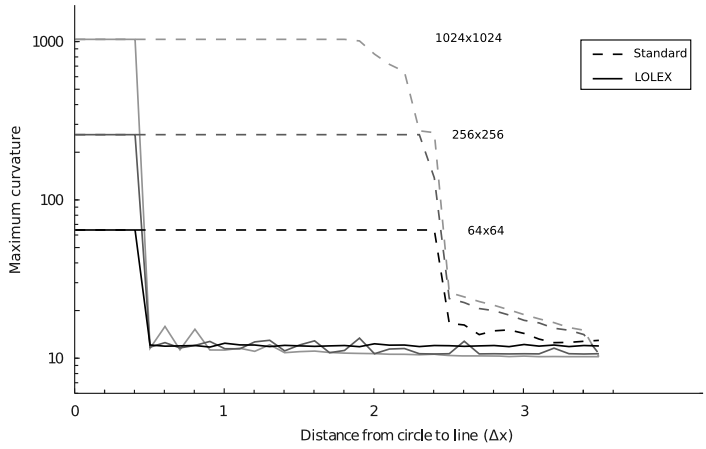



**Fig. 10.** Supremum norm of the curvature for decreasing interface separation. Dashed lines: results using the standard method. Solid lines: results using the LOLEX method. The lines are shaded lighter with increasing grid resolution. The analytical curvature of the circle is 10.

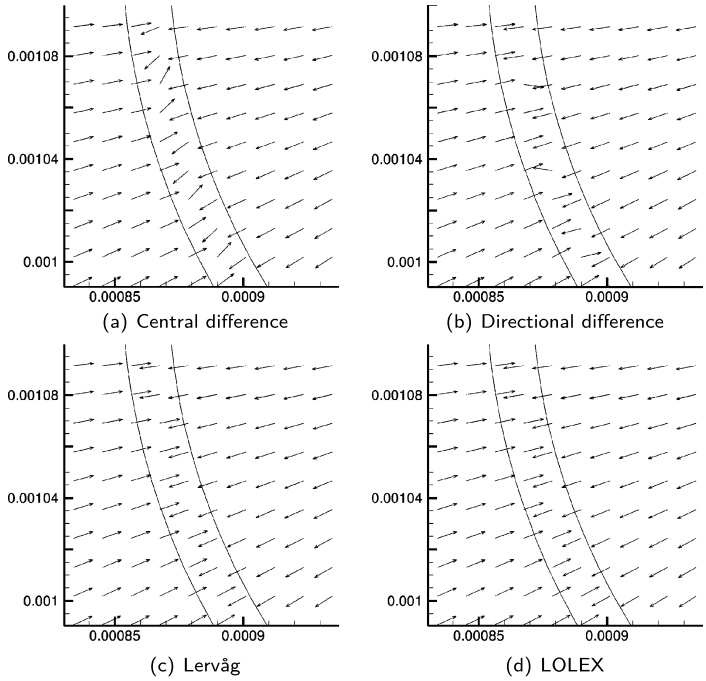



**Fig. 11.** Comparison of normal vector calculations using different methods.

upwind direction. Normal vectors are equally important in calculating the extension velocity, where an error would lead to the interface not moving according to the flow.

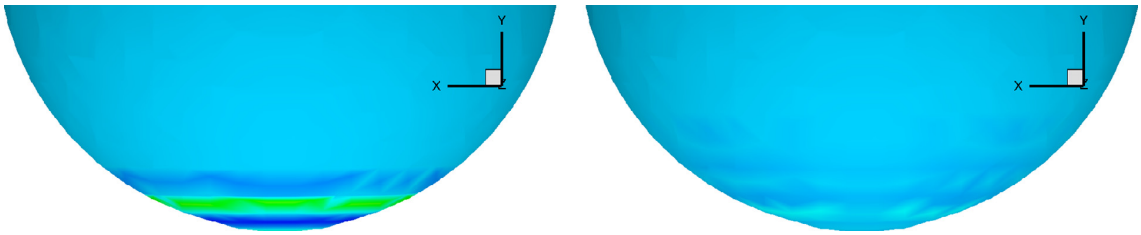
### 4.3. Concentric circles

In order to compare the proposed method to the standard method, a geometric test case was considered which replicates the demands of simulating merging interfaces. The calculated normal vectors are compared both to the standard central-differences discretization, to a directional-differences discretization as described in [4], and to the curve-fitting method of Lervåg [13].

In this test case, two concentric circles were initialized, as if we had a thin ring of fluid 1 inside fluid 2. The width of this ring was $1.6\Delta x$. This test case is interesting, since it reveals grid effects or anisotropies. It also replicates the situation of a thin film that forms between a droplet and a pool for cases where the droplet deforms the pool surface before merging. This has been observed experimentally, see e.g. [36]. The results for all four methods are shown in Fig. 11.

In this figure it is seen that the directional difference method is not much better than the central difference method. This is partly what prompted the use of curve fitting methods; Macklin and Lowengrub initially used directional differences and additional grid refinement in [4], but switched to curve-fitting methods in [10]. As is seen in Fig. 11(c), curve fitting methods (the method by Lervåg is used here) give the correct result. In (d), we see that the LOLEX method also gives the correct result. It is impossible to distinguish the results in (c) and (d) without overlaying the figures and zooming in a lot. We calculated the maximum RMS deviation between the LOLEX method and the other three methods compared in Fig. 11, e.g. $\left\| \sqrt{(n_x(d) - n_x(a))^2 + (n_y(d) - n_y(a))^2} \right\|_\infty$. This was 0.0086 for the Lervåg method, 0.92 for the Central difference and 1.78 for the Directional difference; a 90° difference would give $\sqrt{2}$. We note that the maximum error is largest for the Directional difference, while the average error is largest for the Central difference. The difference between the LOLEX and Lervåg methods is too small to have any impact on the simulation results.

As pointed out several times already, the main advantage of the present LOLEX method over methods employing curve fitting is that it scales easily to 3D. This is because the present method retains the implicitness of the level-set method. A 3D extension of the Macklin and Lowengrub method, on the other hand, would fit a local surface to the point of interest, as they indicate in [10]. Curvature estimation in 3D based on local surface fitting has long been a topic of research in computer vision, see [37] for a review of various methods including the use biquadratic surfaces and of splines. The conclusion of [37] is that these methods are very sensitive to numerical noise (in their context, sensor noise). In the current case, noise is to be expected, as can be seen in Fig. 7(b). Due to this fact, methods in computer vision that avoid local surface fitting and




(a) Standard discretization (b) LOLEX method

**Fig. 12.** 3D bubble above a plane (not shown). Comparison of the standard curvature discretization (a) and the LOLEX method (b). The surfaces are colored according to the curvature, and the standard method is seen to give large errors close to the kink in the level-set function (which is below the sphere), seen as green and dark blue bands. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

calculate only the sign of the curvature have been introduced, since this quantity can be calculated more reliably [38]. This is not a viable alternative in two-phase flow simulations as considered here.

### *4.4. 3D bubble above a plane*

A curvature calculation using the LOLEX method on a 3D case is shown in Fig. 12. In this case, a bubble is placed above a plane, with distance $1.2\Delta x$ at the closest. The grid is $50 \times 50 \times 50$, and the bubble radius is $12.5\Delta x$. The surfaces are colored according to the curvature (interpolated to the surface). In Fig. 12(a), the standard method is used. In 3D, this is the 27-point stencil given by Kang et al. [21]. In Fig. 12(b), the LOLEX method is used to extract the local level sets, and the curvature is then calculated using the same 27-point stencil on these local level sets. It is seen that the LOLEX method performs much better than the standard discretization in areas where the bubble and plane are in close proximity. Note that the plane is not shown here, only the bubble. The kink in the global $\phi$ is below the bubble.

Comparing to the analytical curvature, which in this case is $-10$ for the spherical bubble, it is seen that the standard discretization performs well away from kinks, where the variation in curvature is at most $\pm 0.2\%$. Close to the kink, the standard discretization has errors of $\pm 80\%$, seen as green and dark blue bands in Fig. 12(a). The LOLEX method has the same variation as the standard method away from kinks, while the variation is $\pm 2\%$ close to the kink, seen as light blue spots in Fig. 12(b). Thus it is seen that the LOLEX method gives an error which is an order of magnitude lower than the standard method close to kinks in the level-set function. There is still a small error of the same size as reported above in 2D, which is again probably caused by reinitialization. A deviation of this magnitude is unlikely to have a large impact on simulations, in contrast to the errors from the standard discretization.

To the knowledge of the authors, improved calculation of geometric quantities for a pure level-set formulation in three spatial dimensions that handles general topologies have not been reported before in the literature. Salac and Lu report results of 3D simulations in [11], but it is not known how (or if) they handle problems like that illustrated in Fig. 2, i.e. a body folding back onto itself. They also do not discuss the problem of needing good normal vectors at the interface in order to solve the reinitialization equation.

Given the current state of developments toward petascale supercomputers, and particularly the rapid evolution in GPU-accelerated solvers, dynamic 3D level-set simulations of colliding bodies are going to become more and more common. As this happens, a method such as the present one will be necessary in order to get trustworthy results for situations where accurate curvature is important.

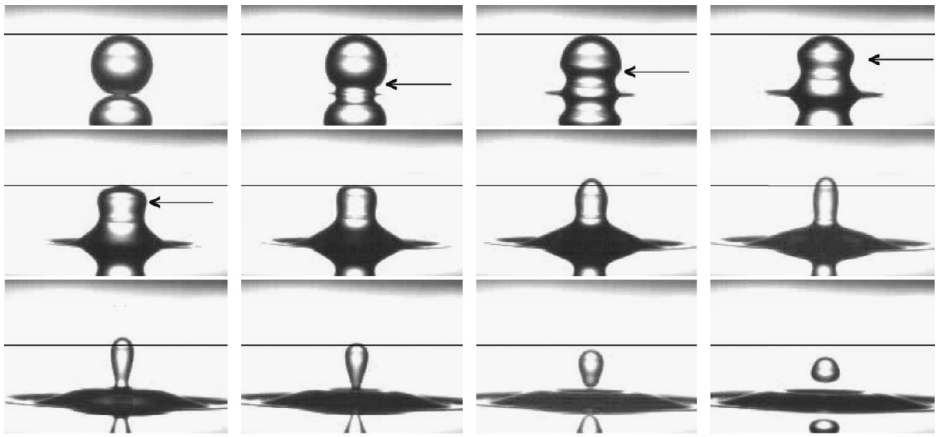
## 5. Dynamic results

As discussed previously, the case of a single droplet of liquid falling onto a pool of the same liquid, either through gas or another liquid, has been widely studied. Thus it is a good benchmark on which to test the proposed method, since detailed experimental results are available.

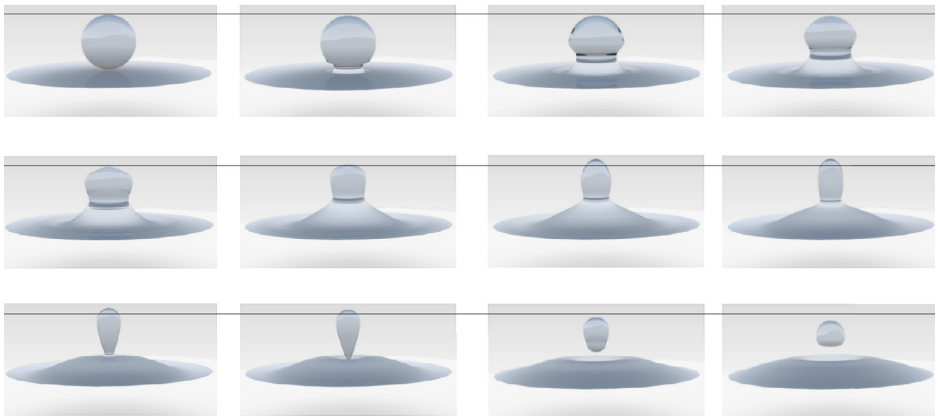
When considering this case, the main dichotomy is between a droplet falling through gas and a droplet falling through liquid. We will consider both cases here, since both are interesting from an industry standpoint. These two cases present different challenges to numerical simulations. The liquid-in-gas case has a high density difference between the two fluids, which is known to be a difficult case. Sussman et al. have studied this problem, and have produced good results using a hybrid LSM-VOF method [18]. The liquid-in-liquid case, on the other hand, can be time-consuming to simulate due to the viscous term in the CFL-criterion used here [21], but is not challenging with respect to density differences.

### *5.1. Decane droplet in water merging with decane pool*

The simulation discussed here consider two immiscible liquids, where a droplet of the heaviest liquid is placed in the lightest liquid above a pool of the heaviest liquid. In the experimental work by Chen et al. [39], the droplet is made to



(a) Experimental result



(b) Simulation result

**Fig. 13.** A 1.1 mm diameter water droplet merging with a water pool. The ambient fluid is 20% polybutene in decane. Snapshots are taken 542 μs apart, the arrow indicates the capillary wave, and the horizontal lines indicate the top of the bubble in the first frame. (a) is the experimental result, reprinted with permission from [39], copyright 2006 American Institute of Physics. (b) is the simulation result.

rest on the pool, and then merging happens after some time. The heavy liquid is water, and the light liquid is a mix of 20% polybutene in decane. The droplet diameter is 1.1 mm. As the droplet and interface are brought into proximity, a thin film is formed between them. This thin film drains, and after some time the film ruptures and the droplet merges with the interface. In the Chen et al. experiments, the merging happens at the central point, but off-center merging has also been reported for larger droplets [40].

A simulation was performed with the same fluid properties and droplet dimension as reported by Chen et al. The computational domain was $6 \times 6$ mm, the grid was $400 \times 400$, and the CFL-number was 0.8. The results are shown in Fig. 13. The agreement between the simulation and experimental results is very good.

In this simulation, the point of merging is decided mainly by grid effects when the droplet deforms the interface forming a thin film. With the present method, we must simply hope that precisely what happens at the time of merging does not significantly affect the following behavior. Comparing Fig. 13(a) and (b) indicates that in this case the precise mechanisms of merging are not very important, as the numerical and experimental results agree very nicely. To accurately capture the thin film behavior, the grid resolution would have to be extremely fine. Hodgson and Lee [41] report that the width of the thin film between a droplet and a pool for the water-toluene system they study is four orders of magnitude smaller than the droplet diameter, i.e. around 100 nanometers. It is possible that an adaptive grid could be able to resolve such a thin film, but since there is no analog to the Knudsen number for liquids, it is not immediately clear whether the continuum description of the Navier–Stokes equations still holds at this length scale.

Comparing the LOLEX method and the standard method on this case, the standard method gives a more oscillatory pressure field around the contact point, as seen in Fig. 14. This increased pressure inside the thin film delays the rupture of the film, seen as a slightly increased width of the film in Fig. 14(b).

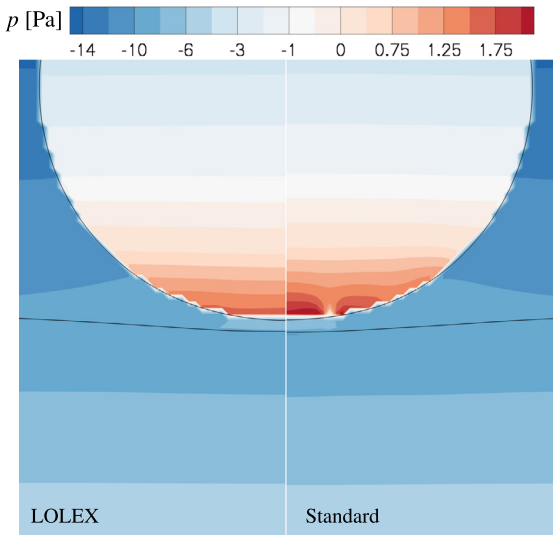



**Fig. 14.** Water droplet in a mixture of polybutene and decane, about to merge with a water pool. Comparison of the pressure field using the LOLEX method and the standard method at $t = 0.007$ s.

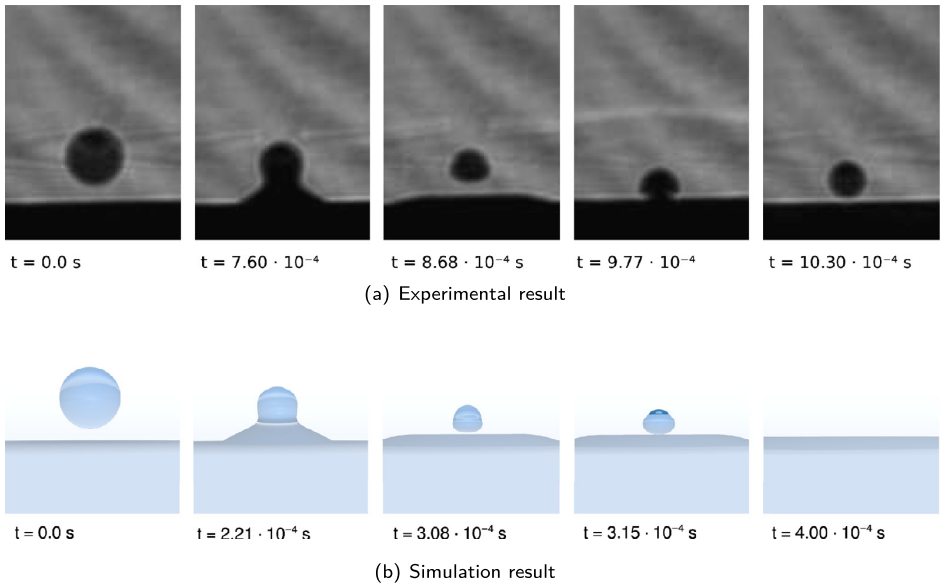
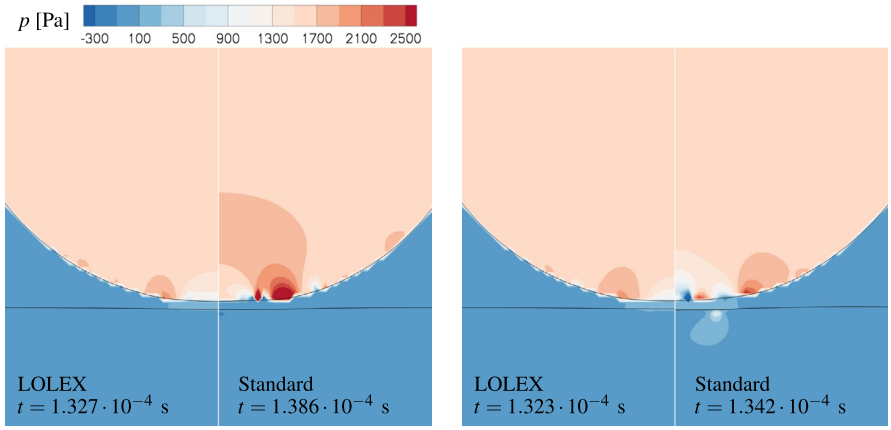



**Fig. 15.** Experimental results (top) and simulation results (bottom) for a 0.18 mm water droplet falling through air and impacting a deep pool of water at 0.29 m/s. (a) is reprinted from [42], Copyright (2011), with permission from Elsevier.

### *5.2. Water droplet falling through air onto a water pool*

Considering the case of a liquid in gas, a simulation was performed of a 0.18 mm diameter water droplet falling through air at 0.29 m/s and impacting a deep pool of water. Experimental results for this case due to Zhao, Brunsvold and Munkejord are found in [42]. These results indicate that a partial coalescence occurs, but the high-speed camera used was not fast enough to capture all the details of the partial coalescence process.

The simulation was performed using axisymmetry. The computational domain was 0.7 mm $\times$ 0.7 mm, resolved using a $401 \times 401$ Cartesian grid. The CFL-number was 0.25. The LOLEX method was used for curvature and normal vector calculation. A comparison of the experimental and simulation results are shown in Fig. 15.




(a) Reinitialization every 7 time steps (b) Reinitialization every time step

**Fig. 16.** Water droplet falling onto a pool, just before the interfaces merge. Comparison between the LOLEX method and the standard method. The pressure field is shown as colored contours.

The time intervals between frames for the experimental and simulation results do not match in this figure. The intervals between the second and third frames are the ones that match best, suggesting that the behavior of the thin air film that forms between the droplet and the pool before coalescence is the major source of this discrepancy. The grid used in the simulation is unable to resolve the thin film. It is not clear that an increased grid resolution would amend this, as the continuum approximation may not be valid for the thin air film. The width of this film is not known from experiments.

As an order-of-magnitude estimate, we can use the results by Hodgson and Lee [41]. They report that the width of the thin film between a droplet and a pool before merging, for the water-toluene system they study, is around $L = 100$ nanometers. Since the mean free path in air at room temperature and atmospheric pressure is around $\lambda = 66$ nanometers [43], the Knudsen number is $Kn = \frac{\lambda}{L} \approx 0.7 \ll 1$, which would imply that the continuum description is no longer valid.

Nevertheless, the simulation is able to correctly predict the partial coalescence, and the simulation agrees well with experiments on the size of the daughter droplet produced. In the experiments, this daughter droplet subsequently bounces on the pool of water. The simulation is unable to predict this, again due to the thin air film formed, and shows the daughter droplet merging with the water pool instead.

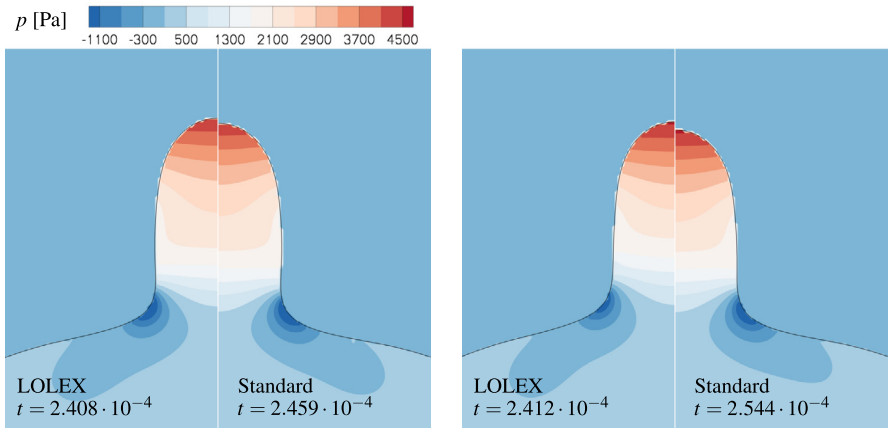
A comparison between the LOLEX method and the standard method is shown in Figs. 16 and 17. These figures show a section through the droplets just before collision and just when the neck is at its tallest, respectively. The pressure field is plotted as colored contours. The LOLEX method is plotted on the left side and the standard method is plotted on the right side. It is seen from these figures that the curvature errors produced by the standard method give rise to significant oscillations in the pressure; note in particular the interleaved red and blue patches where the pressure changes sign. As the reinitialization is performed more frequently, the oscillations persist, and are even found inside the pool below the droplet.

An important effect of this erroneous pressure is a loss of kinetic energy, which can be seen in Fig. 17, where the neck is clearly shorter with the standard method. It is also seen that more frequent reinitialization leads to a higher loss of kinetic energy. As some authors have noted [44], the height of the neck and the dynamics of the capillary waves are important factors for the partial coalescence mechanism.

The LOLEX method is not significantly affected by the amount of reinitialization, and gives a more sensible pressure field in both cases. It should be noted that the pressure difference across the droplet interface in Fig. 16 is about 2500 Pa, which is very large, caused by the very small droplet diameter.

This case also allows an illustration of the benefit of using the LOLEX method over the Salac and Lu method. In Fig. 18 we compare the curvature field for these two just after droplet-film merging, where it is seen that curvature errors in the Salac and Lu method have led to the entrainment of a small bubble. Since the bubble is under-resolved on this grid, it subsequently disappears due to reinitialization. The LOLEX method does not entrain any bubbles.

Finally, we consider the performance impact of the LOLEX method on this case. The same simulations using the LOLEX and standard method were rerun using a $201 \times 201$ grid for timing purposes. The simulation using the standard method took 43 525 s of CPU time, while the simulation using the LOLEX method took 46 753 s. This means the LOLEX method is 7% slower than the standard method for this case, which is a fair trade-off for the benefits of both reduced pressure oscillations and lower sensitivity to reinitialization frequency.




(a) Reinitialization every 7 time steps (b) Reinitialization every time step

**Fig. 17.** Water droplet falling onto a pool, when the neck reaches its highest position. Comparison between the LOLEX method and the standard method. The pressure field is shown as colored contours.

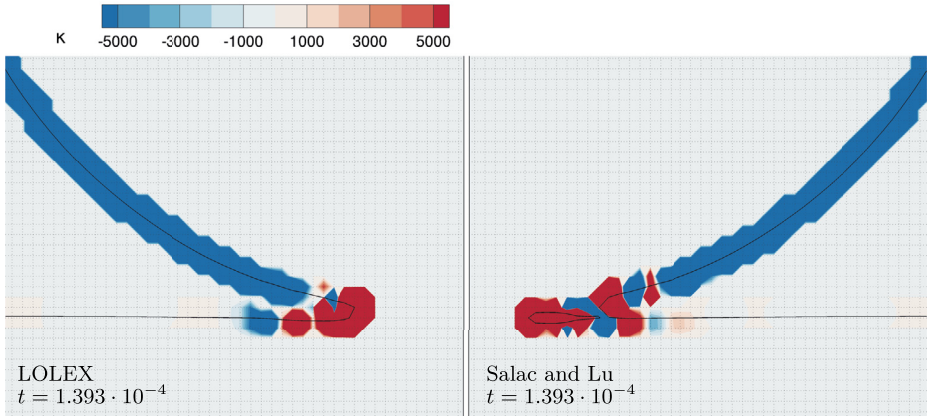



**Fig. 18.** Water droplet falling onto a pool, zoom in on the interface just after merging. Comparison between the LOLEX method and the Salac and Lu method. It is seen that the latter entrains a small air bubble due to the oscillatory curvature field following the merging. The curvature field is shown as colored contours.

## 6. Concluding remarks

In the present work we have proposed a new method for calculating the curvature and normal vectors of an interface represented by a level-set function, and which gives accurate results before, during and after topological changes in the interface. The method is compared to the standard method for geometric test cases, where the analytical curvature is known, and it is seen that in areas where the standard method gives errors of around 100%, the proposed method gives errors of around 1–2%. The method is easily extended to 3D, as is demonstrated, where the same reduction in error is seen. The method is then employed in simulations of two-phase flow where a droplet merges with a pool. Here it is seen that the standard method gives rise to unphysical pressure oscillations before merging, which affect the subsequent capillary waves, while the proposed method fares much better. The results of the simulations using the proposed method are compared to experimental results both for a liquid-in-liquid case, where the agreement is very good, and for a more demanding liquid-in-gas case where the agreement is qualitative, reproducing the partial coalescence behavior.

## Acknowledgements

This work was financed through the Enabling Low-Emission LNG Systems project at SINTEF Energy Research, and the authors acknowledge the contributions of GDF SUEZ, Statoil and the Petromaks programme of the Research Council of

Norway (193062/S60). We also thank one of the anonymous reviewers for detailed and extensive comments which helped improve the quality of this work.

## References

[1] L. Rayleigh, Proc. R. Soc. 28 (1878) 405–409.
[2] A.M. Worthington, Proc. R. Soc. 25 (1876) 261–272.
[3] S.T. Thoroddsen, K. Takehara, T.G. Etoh, S. Popinet, P. Ray, C. Josserand, S. Zaleski, M.-J. Thoraval, von Kármán vortex street within an impacting drop, Phys. Rev. Lett. 108 (2012) 264506.
[4] P. Macklin, J. Lowengrub, Evolving interfaces via gradients of geometry-dependent interior Poisson problems: application to tumor growth, J. Comput. Phys. 203 (2005) 191–220.
[5] V. Mallet, D. Keyes, F. Fendell, Modeling wildland fire propagation with level set methods, Comput. Math. Appl. 57 (2009) 1089–1101.
[6] V. Melicher, I. Cimrak, R.V. Keer, Level set method for optimal shape design of MRAM core. Micromagnetic approach, Physica B, Condens. Matter 403 (2008) 308–311.
[7] S. Osher, R.P. Fedkiw, Level set methods: An overview and some recent results, J. Comput. Phys. 169 (2001) 463–502.
[8] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[9] P. Smereka, Semi-implicit level set methods for curvature and surface diffusion motion, J. Sci. Comput. 19 (2003) 439–456.
[10] P. Macklin, J. Lowengrub, An improved geometry-aware curvature discretization for level set methods: Application to tumor growth, J. Comput. Phys. 215 (2006) 392–401.
[11] D. Salac, W. Lu, A local semi-implicit level-set method for interface motion, J. Sci. Comput. 35 (2008) 330–349.
[12] K.Y. Lervåg, Calculation of interface curvature with the level-set method, in: Sixth National Conference on Computational Mechanics MekIT'11, Trondheim, Norway, 2011.
[13] K.Y. Lervåg, B. Müller, S.T. Munkejord, Calculation of the interface curvature and normal vector with the level-set method, Comput. Fluids 84 (2013) 218–230.
[14] Å. Ervik, The local level-set extraction method for robust calculation of geometric quantities in the level-set method, Master's thesis, Norwegian University of Science and Technology (NTNU), 2012.
[15] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.
[16] D. Hartmann, M. Meinke, W. Schröder, The constrained reinitialization equation for level set methods, J. Comput. Phys. 229 (2010) 1514–1535.
[17] J.E. Pilliod Jr., E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, J. Comput. Phys. 199 (2004) 465–502.
[18] M. Sussman, K. Smith, M. Hussaini, M. Ohta, R. Zhi-Wei, A sharp interface method for incompressible two-phase flows, J. Comput. Phys. 221 (2007) 469–505.
[19] S. Shin, I. Yoon, D. Juric, The local front reconstruction method for direct simulation of two- and three-dimensional multiphase flows, J. Comput. Phys. 230 (2011) 6605–6646.
[20] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A PDE-based fast local level set method, J. Comput. Phys. 155 (1999) 410–438.
[21] M. Kang, R.P. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (2000) 323–360.
[22] E.B. Hansen, Numerical simulation of droplet dynamics in the presence of an electric field, Ph.D. thesis, Norwegian University of Science and Technology (NTNU), 2005.
[23] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[24] R.P. Fedkiw, X.D. Liu, The Ghost Fluid Method for viscous flows, presented at the "Solutions of PDE", conference in honour of Prof. Phil Roe, 1998.
[25] A. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.
[26] S. Balay, J. Brown, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc web page, http://www.mcs.anl.gov/petsc, 2012.
[27] J. Kraaijevanger, Contractivity of Runge–Kutta methods, BIT Numer. Math. 31 (1991) 482–528.
[28] D.I. Ketcheson, A.C. Robinson, On the practical importance of the SSP property for Runge–Kutta time integrators for some common Godunov-type schemes, Int. J. Numer. Methods Fluids 48 (2005) 271–303.
[29] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
[30] K.Y. Lervåg, Å. Ervik, Curvature calculations for the level-set method, in: ENUMATH 2011 Proceedings Volume, Leicester, England.
[31] C. Focke, D. Bothe, Direct numerical simulation of binary off-center collisions of shear thinning droplets at high Weber numbers, Phys. Fluids 24 (2012) 73105.
[32] P. Trontin, S. Vincent, J. Estivalezes, J. Caltagirone, A subgrid computation of the curvature by a particle/level-set method. Application to a front-tracking/ghost-fluid method for incompressible flows, J. Comput. Phys. 231 (2012) 6990–7010.
[33] P. Verbeek, L. van Vliet, J. van de Weijer, Improved curvature and anisotropy estimation for curved line bundles, in: A.K. Jain, S. Venkatesh, B.C. Lovell (Eds.), Fourteenth International Conference on Pattern Recognition, vols. 1 and 2, International Conference on Pattern Recognition, 1998, pp. 528–533.
[34] D. Adalsteinsson, J.A. Sethian, A fast level set method for propagating interfaces, J. Comput. Phys. 118 (1995) 269–277.
[35] D. Adalsteinsson, J.A. Sethian, The fast construction of extension velocities in level set methods, J. Comput. Phys. 148 (1999) 2–22.
[36] Z. Mohamed-Kassim, E.K. Longmire, Drop impact on a liquid–liquid interface, Phys. Fluids 15 (2003) 3263–3273.
[37] P. Flynn, A. Jain, On reliable curvature estimation, in: Computer Vision and Pattern Recognition, Proceedings CVPR '89, in: IEEE Comput. Soc. Conf., 1989, pp. 110–116.
[38] C.-K. Tang, G. Medioni, Curvature-augmented tensor voting for shape inference from noisy 3D data, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 858–864.
[39] X. Chen, S. Mandre, J.J. Feng, Partial coalescence between a drop and a liquid–liquid interface, Phys. Fluids 18 (2006) 051705.
[40] Z. Mohamed-Kassim, E.K. Longmire, Drop coalescence through a liquid/liquid interface, Phys. Fluids 16 (2004) 2170–2181.
[41] T. Hodgson, J. Lee, The effect of surfactants on the coalescence of a drop at an interface i, J. Colloid Interface Sci. 30 (1969) 94–108.
[42] H. Zhao, A. Brunsvold, S.T. Munkejord, Transition between coalescence and bouncing of droplets on a deep liquid pool, Int. J. Multiph. Flow 37 (2011) 1109–1119.
[43] S.G. Jennings, The mean free path in air, J. Aerosol Sci. 19 (1988) 159–166.
[44] F. Blanchette, T.P. Bigioni, Partial coalescence of drops at liquid interfaces, Nat. Phys. 2 (2006) 254–257.

# Paper 2

# EXTENDING A SERIAL 3D TWO-PHASE CFD CODE TO PARALLEL EXECUTION OVER MPI BY USING THE PETSC LIBRARY FOR DOMAIN DECOMPOSITION

**Åsmund ERVIK**[1*], **Svend Tollak MUNKEJORD**[2†], **Bernhard MÜLLER**[1‡]

[1]NTNU Department of Energy and Process Engineering, NO-7491 Trondheim, NORWAY

[2]SINTEF Energy Research, P.O. Box 4761 Sluppen, NO-7465 Trondheim, NORWAY

* E-mail: asmund.ervik@ntnu.no
† E-mail: svend.t.munkejord@sintef.no
‡ E-mail: bernhard.muller@ntnu.no

## ABSTRACT

To leverage the last two decades' transition in High-Performance Computing (HPC) towards clusters of compute nodes bound together with fast interconnects, a modern scalable CFD code must be able to efficiently distribute work amongst several nodes using the Message Passing Interface (MPI). MPI can enable very large simulations running on very large clusters, but it is necessary that the bulk of the CFD code be written with MPI in mind, an obstacle to parallelizing an existing serial code.

In this work we present the results of extending an existing two-phase 3D Navier-Stokes solver, which was completely serial, to a parallel execution model using MPI. The 3D Navier-Stokes equations for two immiscible incompressible fluids are solved by the continuum surface force method, while the location of the interface is determined by the level-set method.

We employ the Portable Extensible Toolkit for Scientific Computing (PETSc) for domain decomposition (DD) in a framework where only a fraction of the code needs to be altered. We study the strong and weak scaling of the resulting code. Cases are studied that are relevant to the fundamental understanding of oil/water separation in electrocoalescers.

**Keywords:** Parallelization, Oil/water separation, Surfactants and interfaces, Bubble and droplet dynamics .

## NOMENCLATURE

| | | |
|---|---|---|
| $\mu$ | Dynamic viscosity of a fluid. | Pa·s |
| $\nu$ | Kinematic viscosity of a fluid. | $m^2/s$ |
| $\rho$ | Density of a fluid. | $kg/m^3$ |
| $\mathbf{f}$ | External acceleration. | $m/s^2$ |
| $\mathbf{u}(\mathbf{x})$ | Velocity field of a fluid. | m/s |
| $p(\mathbf{x})$ | Pressure of a fluid. | Pa |
| $\kappa$ | Curvature of the interface. | 1/m |
| $\sigma$ | Coefficient of surface tension. | N/m |
| $n$ | Time step index. | |
| Re | Reynolds number. | |

## INTRODUCTION

In 1965 Gordon Moore famously predicted that transistor density (and hence computing power for a given chip) would double each year in the foreseeable future (Moore, 1965). Dubbed Moore's law, this trend continued to hold for roughly 40 years and meant that life was easy for people needing greater and greater computational power. While serious High-Performance Computing (HPC) was dominated in most of this period by vector machines like the seminal Cray 1, by the mid-1990s clusters of many interconnected scalar CPUs had become a cheaper solution, leading to the industry-wide adoption of distributed memory architectures.

Around 2005 Moore's law finally started hitting a barrier when the high heat production of chips and, somewhat later, the diffraction limits for photolitography began forcing chip makers to alter their ways. Two complementary solutions were introduced, namely shared-memory architectures (multi-core CPUs) and vector instruction sets (SSE, AVX, FMA)[1]. Both solutions were adopted in HPC, leading to hybrid shared-memory/distributed-memory systems. In the last five years accelerator technologies (GPGPU, MIC)[2] have furthered the return to vector processing, so HPC has in a sense come full circle. All in all this gives a very heterogeneous environment for HPC where the onus is on the application programmer to ensure that his/her code can make the most of the available resources.

In contemporary numerical codes, omitting here the use of accelerators, the two main programming paradigms for leveraging parallelism are OpenMP and MPI. OpenMP takes advantage of shared-memory architectures, while MPI can use distributed-memory architectures. On current systems, OpenMP can scale from 1 to 32 cores, while MPI can scale to thousands and even millions of cores. This means that MPI is the paradigm of choice for HPC, possibly in combination with OpenMP used by each MPI process.

We will use the following nomenclature when discussing parallelism: a "process" is one MPI rank which is executing code. A CPU has several "cores", each of which may execute a process. The CPUs are located on "nodes", e.g. a desktop computer or a blade in a cluster. Typical cluster nodes have 2 (or more) CPUs, each having a separate "socket" connecting the CPU to

[1]SSE: Streaming SIMD Extensions. AVX: Advanced Vector Extensions. FMA: Fused Multiply-Add.

[2]GPGPU: General-Purpose Graphics Processing Unit. MIC: Many Integrated Core.

the memory (RAM). Each socket has one communication channel to memory shared by all cores on this socket. Many nodes can communicate over the "interconnect", which should preferrably be very fast and have very low latency.

This paper will focus on the use of MPI to port an existing serial implementation of a 2D/3D incompressible Navier-Stokes solver. This code can simulate two-phase flows relevant e.g. for the fundamental understanding of oil/water separation, but for 3D cases the runtime is very long (weeks and months). The majority of this runtime is due to the solution of a Poisson equation for the pressure, and state of the art algorithms for this problem are bound by the memory bandwidth rather than CPU speed. This makes OpenMP a poor solution in this case and leaves MPI as the necessary paradigm for parallelism. We will employ the PETSc library, specifically the DMDA component, to do domain decomposition. The solution of the Poisson equation is also done using PETSc routines. We establish a framework in Fortran where it is possible to reuse the existing serial code.

The rest of this paper is organized as follows: in the next section, the basic equations are established, after which the numerical methods are presented. Then we describe the framework and the specific changes that were needed to port the serial code. Computations performed with the resulting code are discussed and we study the strong and weak scaling on several architectures. Finally some closing remarks are given.

## MODEL DESCRIPTION

The equations that govern the two-phase flow system under consideration are the incompressible Navier-Stokes equations:

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0 \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \boldsymbol{\nabla})\mathbf{u} = -\frac{\boldsymbol{\nabla} p}{\rho} + \frac{\mu}{\rho}\boldsymbol{\nabla}^2 \mathbf{u} + \mathbf{f} \tag{2}$$

These equations are valid for single-phase flow. To extend this formulation to two-phase flow we keep these equations in each of the two phases, where the densities and viscosities are constant in each phase. We will restrict ourselves to laminar flow, as we are interested in situations with Reynolds numbers $\mathrm{Re} \sim \mathcal{O}(1)$.

Across the interface between the fluids, a jump in the normal component of the traction vector will arise due to the surface tension $\sigma$, and this jump together with effects of the jump in density and viscosity must be added to our equations. We introduce these effects using the continuum surface force method (CSF) (Brackbill *et al.*, 1992). The location of the interface is captured using the level-set method (LSM) (Osher and Sethian, 1988; Osher and Fedkiw, 2001), see Ervik *et al.* (2014) for a detailed description, we provide only a short outline here.

The level-set method is a method for capturing the location of an interface. It is widely used not just for multi-phase fluid flow but also in other contexts where an interface separates two regions. The interface is represented by a level-set function $\phi(\mathbf{x})$ which is equal to the signed distance to the interface. In other words, the interface is given by the zero level set $\{\mathbf{x} \,|\, \phi(\mathbf{x}) = 0\}$, hence the name. Rather than advecting the interface location, one advects the function $\phi(\mathbf{x})$ directly according to the transport equation

$$\frac{\partial \phi}{\partial t} = -\mathbf{u} \cdot \boldsymbol{\nabla} \phi \tag{3}$$

giving an implicit formulation that automatically handles changes in interface topology.

The level-set method can be visualized as in Fig. 1 for a 2D fluid flow with a drop next to a film, seen on the right-hand side in this figure as gray shapes. The distance is shown as isocontour lines superimposed on these shapes. On the left-hand side the level-set function is shown visualized in 3D as surfaces where the height above water corresponds to the signed distance. The analogy to a map describing an island rising out of the water is quite striking, except that the roles of "reality" and "tool for description" have been reversed.

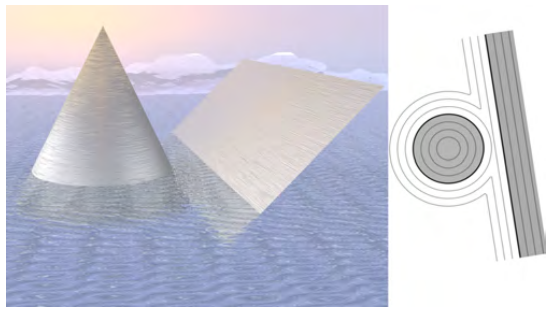


Figure 1: Illustration of the level-set method. Right: in 2D, a fluid drop (dark gray) seen next to a fluid film (dark gray), both immersed in a different fluid (white). Left: the signed-distance function representing these two fluid bodies, the drop and the film.

When the location of the interface is known, the curvature $\kappa$ can be calculated from $\phi$, and together with $\sigma$ this gives the surface tension force. In the CSF method this force is incorporated as a volume-force term which is non-zero only in a thin band around the interface. This thin band is produced by smearing out the delta function, making the force term continuous For such a smeared-out delta function, we can compute the volume-force term at a point $\mathbf{x}$ close to the interface as

$$\mathbf{f}_\mathrm{s}(\mathbf{x}, t) = \int_\Gamma \mathbf{f}_\mathrm{sfd}(\mathbf{s}, t)\delta(\mathbf{x} - \mathbf{x}_I(s))d\mathbf{s}, \tag{4}$$

where $\mathbf{f}_\mathrm{sfd}$ is a surface-force density and $\mathbf{x}_I(s)$ is a parametrization of the interface. The surface-force density is such that the integral of $\mathbf{f}_\mathrm{s}(\mathbf{x}, t)$ across the (smeared-out) interface approximates the surface tension force, see Brackbill *et al.* (1992) for details. Note that in the level-set context it is not necessary to parametrize the interface since $\phi(\mathbf{x})$ stores the distance to the interface, so we have $\mathbf{x} - \mathbf{x}_I(s) = \phi(\mathbf{x})$ as long as $\phi(\mathbf{x})$ is a signed distance function. There are several ways to smear out the delta function, we follow Osher

and Fedkiw (2003, Eq. 1.23),

$$\delta(x) = \begin{cases} 0 & \text{if } |\phi| > \epsilon \\ \frac{1}{2\epsilon}\left(1 + \cos\left(\frac{\pi\phi}{\epsilon}\right)\right) & \text{else} \end{cases} \tag{5}$$

where $\epsilon = 1.5\Delta x$ is employed. This one-dimensional delta function is composed into the three-dimensional version by taking $\delta(\mathbf{x}) = \delta(x)\delta(y)\delta(z)$.

This formulation leads to a source term which incorporates the effects of surface tension. It is also necessary to smear out the viscosity and density differences across the interface in order to be consistent with the above formulation. A smeared-out Heaviside function $H(\mathbf{x})$ is used to accomplish this, given by Osher and Fedkiw (2003, Eq. 1.22) as

$$H(x) = \begin{cases} 0 & \text{if } \phi < -\epsilon \\ \frac{1}{2}\left(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin\left(\frac{\pi\phi}{\epsilon}\right)\right) & \text{if } -\epsilon < \phi < \epsilon \\ 1 & \text{if } \phi > \epsilon \end{cases} \tag{6}$$

## NUMERICAL METHODS

To discretize the Navier-Stokes equations and the equations for the level-set method we employ finite difference methods, specifically WENO (Liu *et al.*, 1994) for the convective terms and central differences for the viscous terms in Eq. (2), and WENO also for Eq. (3). The time integration is done with an explicit second-order Runge-Kutta method (SSPRK (2,2) in the terminology of Gottlieb *et al.* (2009)) for both Eq. (2) and Eq. (3).

The grid is a structured rectangular uniform staggered grid. A staggered grid is employed to avoid checkerboarding of the pressure field; this means that the pressure and other scalars "live" at cell centers, while the velocities "live" at the cell faces. To be more precise, if we have a pressure at one point $p_{i,j,k}$, the velocities $(u, v, w)$ around this point are $u_{i\pm1/2,j,k}, v_{i,j\pm1/2,k}, w_{i,j,k\pm1/2}$ located at the 6 cell faces. In the actual code we store the velocity values for $u_{i+1/2,j,k}$, $v_{i,j+1/2,k}$, $w_{i,j,k+1/2}$ at the index `(i,j,k)` even though these values are not physically colocated.

The major problem when solving Eqs. (1) and (2) is that this is not a set of PDEs, it is a differential-algebraic equation (DAE) with a Hessenberg index of two. In other words, even though we have four equations (Eq. (2) is three equations) and four unknowns $(u, v, w, p)$, Eq. (1) cannot be used directly to find $p$. The first solution to this conundrum was presented by Chorin (1968). This method can be understood as calculating an approximate velocity field $\mathbf{u}^*$ which does not satisfy Eq. (1), and subsequently projecting this velocity field onto the manifold of vector fields satisfying Eq. (1). For this reason, the method is often called Chorin's projection method or simply *the* projection method. It consists of these three steps, where we calculate three quantities successively, namely $\mathbf{u}^*, p_{n+1}, \mathbf{u}_{n+1}$:

$$\frac{\mathbf{u}^* - \mathbf{u}_n}{\Delta t} = -\left(\mathbf{u}_n \cdot \boldsymbol{\nabla}\right)\mathbf{u}_n + \nu\boldsymbol{\nabla}^2\mathbf{u}_n \tag{7}$$

$$\boldsymbol{\nabla}^2 p_{n+1} = \frac{\rho}{\Delta t}\boldsymbol{\nabla}\cdot\mathbf{u}^* \tag{8}$$

$$\mathbf{u}_{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho}\boldsymbol{\nabla}p_{n+1} \tag{9}$$

The pressure Poisson equation (8) that arises here is elliptic, so the numerical solution is very time consuming and a vast amount of research has gone into developing fast solvers. For two-phase flows with high density differences, the condition number of the matrix that results when Eq. (8) is discretized will make matters even worse than for the single-phase problem (Duffy *et al.*, 2002). This matrix is very large even in sparse storage formats, for a $256^3$ grid it has 117 million non-zero elements. The current state-of-the-art consists in combining a (geometric or algebraic) multigrid preconditioner with a conjugate gradient method (often BiCGStab) for solving the resulting sparse linear system. Our experience with 2D axisymmetric simulations suggests that the Bi-Conjugate Gradient Stabilized method (van der Vorst, 1992) with the BoomerAMG preconditioner (Henson and Yang, 2000) is an optimal choice. For the simulations performed here, however, the straigth-forward successive over-relaxation (SOR) preconditioner turned out to be faster than BoomerAMG. This has not been investigated in greater detail. We employ the PETSc and Hypre libraries for these methods (Balay *et al.*, 2014; *hypre* , 2014).

We note also that the boundary conditions for Eq. (8) are of pure Neumann type (unless e.g. an outlet pressure is specified), which results in a singular matrix. These boundary conditions arise from the projection method and are not physical. The common "engineering" approach of fixing the singularity, simply fixing the pressure at some point in the domain, is not a very good approach as it may pollute the spectrum of the preconditioner. Instead, projecting the discretized singular equation into the orthogonal complement of the null space of the singular matrix seems to be a good solution (Zhuang and Sun, 2001). In other words, for $Ax = b$, we construct the Krylov operator $K = (\mathbb{I} - \mathbb{N})P^{-1}A$ such that $b, Kb, K^2b, ...$ is orthogonal to the null space $\mathbb{N}$. Here $\mathbb{I}$ is the identity matrix, so $(\mathbb{I} - \mathbb{N})P^{-1}$ is the desired projection. In the PETSc library that we employ here (Balay *et al.*, 2014), this is achieved using the `KSPSetNullSpace()` routine.

## PARALLELIZATION

The starting point for the parallelization was an in-house code consisting of a 2D/3D Navier-Stokes solver and a multi-physics framework that enables the simulation of two-phase flows with the possibility of applying electric fields, and/or adding surface-active agents to the interface. The interface between the two phases is captured using a level-set method, so interfaces with changing topology such as two merging drops can be simulated. The code has been successfully applied to the study of both liquid-liquid (Teigen and Munkejord,

2010) and liquid-gas systems (Ervik *et al.*, 2014), but the long runtime has restricted its use to 2D axisymmetric cases so far.

The PETSc DMDA framework for domain decomposition was chosen as the main methodology for parallelizing the code. Domain decomposition consists in splitting the whole domain into subdomains which are each distributed to one MPI node. Each node then has a computational domain with some internal cells where the flow is computed, and some ghost cells which represent either boundary conditions or values that belong to neighbouring domains. This means that regular communication between the nodes is necessary such that all ghost cells have correct values. Such a splitting is shown in Fig. 2 below. Neglecting for a moment the pressure Poisson equation, this approach can scale well to millions of CPU cores, see e.g. Rossinelli *et al.* (2013) for an example in compressible flow.

By using the PETSc DMDA framework we can avoid the gritty details of domain decomposition and MPI programming. At the initialization of the code, some routines are called to set up three DMDAs which are objects that manage the decomposition. Using these objects we input how large our computational domain should be in terms of grid points, and the library decides an optimal decomposition *at runtime* depending on the number of MPI processes the code is run with. We also specify the physical dimensions of our uniform grid, and the library returns the physical dimensions for each subdomain.

This framework is very convenient, but one enhancement was made to further facilitate the reuse of the serial code. In the standard PETSc framework, the local work arrays that represent the solution on a given subdomain and the values in the ghost cells are indexed using the global indices. The existing code naturally expects indices that go from 1 to the maximum value `imax`. In Fortran, the bounds of an array may be re-mapped when the array is passed to a subroutine, and this feature was used to ensure that each local work array had bounds as expected by the serial code. Thus we will use `imax` as the final `i` index *on each subdomain* in the following.

With this enhancement, the only thing that had to be re-written in the original code was the handling of the staggered grid for the velocity. In the formulation used here, we have one less point for e.g. $u$ in the $x$ direction, since these values are located at the cell faces. In the serial code this is handled by not solving for $u$ at the point `imax`. In the parallel version, $u$ at the point `imax` should however be solved for on those processes that are not at the actual boundary but where there is a neighbouring process in the positive $x$-direction.

Furthermore, this means that a communication step is also necessary in the projection method. After the pressure has been calculated from the Poisson equation, we calculate e.g. the $x$-component of $\boldsymbol{\nabla} p$ at the cell face corresponding to $u$ at `imax`. Numerically this is `(p(imax+1) - p(imax))/dx`, so the ghost value at `imax+1` must be updated before this calculation for those subdomains where `p(imax+1)` represents a pressure value on another subdomain and not a boundary condition.

Returning to the pressure Poisson equation Eq. (8), the elliptic nature of this equation means that, in some sense, all nodes must communicate during the solution. A further reduction in speedup potential is due to the fact that the solvers for this equation are mostly bound by memory bandwidth, which is shared amongst all cores on a modern CPU. These limits imply that we must lower our expectations somewhat in comparison with the impressive results mentioned earlier for compressible flows.

In the DMDA framework, the Poisson equation is set up such that each process computes its own portion of the matrix and right-hand side vector. This is the only scalable way of solving it, even when sparse storage formats are used.

## RESULTS

### Manufactured solution case

After the code had been parallelized it was tested using a manufactured solution (Roache, 2002) inspired by that used in John *et al.* (2006). The debugging tool Valgrind (Nethercote and Seward, 2007; Nethercote *et al.*, 2014) was used in the memory checking mode to ensure that the code does not e.g. make use of uninitialized values, a common programming error. When all such errors were fixed, the code was used to solve the single-phase Navier-Stokes equations with the following exact solution used as an initial – boundary value problem on a (1.0 m)$^3$ domain, where the origin is in the lower left front corner (cf. Fig. 2).

$$\begin{aligned} u &= t^3 yz \\ v &= t^2 xz \\ w &= txy \\ p &= x + y + z - 1.5 \end{aligned} \tag{10}$$

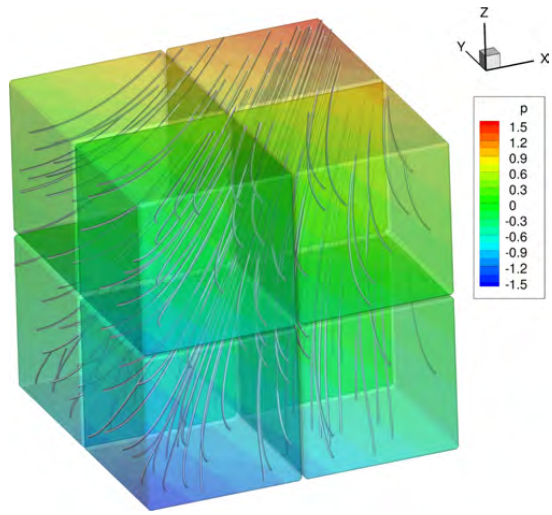



Figure 2: The computed solution after 0.031 s (100 time steps) on a $128^3$ grid run on 8 processors. The blocks show the decomposition of the domain, the pressure field is shown superimposed on these blocks, and the streamlines illustrate the flow.

Insertion into Eq. (1) confirms that this solution is divergence free, and the resulting body force can be computed by inserting Eq. (10) in Eq. (2). In order to minimize the risk of human error, this was done symbolically using Maple, the resulting expression was copied into the Fortran code and regular expressions were used to convert Maple syntax into Fortran. A plot of the computed solution is shown in Fig. 2. Here the velocity streamlines are shown together with the pressure field which has been superimposed on blocks representing the parallel decomposition.

### Convergence

Using the manufactured solution in Eq. (10), the convergence under grid- and time step refinement, as well as the strong and weak scaling, was tested on the Kongull cluster at NTNU. This cluster has dual-socket nodes with Intel Xeon E5-2670 8-core CPUs and a 1 Gb/s Ethernet interconnect. The STREAM benchmark (McCalpin, 2014, 1995) was run on one core and gave an effective memory bandwidth of 9800 MB/s for the Triad test[3].

To test the grid- and time step refinement, a base case was selected with a $256^3$ grid, giving a grid spacing `dx` of $3.91 \cdot 10^{-3}$ m, the CFL condition following Kang *et al.* (2000) with a CFL number of 0.5 then giving a time step of $1.28 \cdot 10^{-4}$ s. This case was solved for 100 time steps, as were solutions on coarser grids $128^3$ and $64^3$ computed with the same time step. All simulations were run on 32 processes (8 nodes with 4 processes each). Subsequently, the same cases were run but with $1/2$, $1/4$ and $1/8$ the time step using 200, 400 and 800 time steps, respectively. The results are shown in Fig. 3.

It is seen that the convergence behaviour is as expected. First of all, the temporal order is 1 (not 2) due to an irreducible splitting error from the projection method. This can be overcome e.g. by using the incremental pressure form (see Guermond *et al.* (2006) for a review of projection methods), but has not been considered in this work. Second, the grid refinement does not influence the error. This is due to the fact that the velocity field is linear in space, so the error is completely dominated by the temporal order.

### Strong scaling

To test the strong scaling of our code, i.e. how simulating a given case speeds up when more processes are used, a $128^3$ grid was used giving a grid spacing `dx` of $7.81 \cdot 10^{-3}$ m, the CFL condition giving a time step of $3.10 \cdot 10^{-4}$ s. The solution was computed for 100 time steps. Since the Poisson solver performance should be bound by memory bandwidth, the test was made using 2 processes per node (one per socket) and several nodes. The resulting speedup relative to one process is shown in Fig. 4. In this figure, the black points indicate the speedup compared to running on one process. The scaling seen is quite good, but as expected lower than the theoretical linear scaling. It is seen that the peak memory usage (orange) increases slightly with more processes.

[3]The Triad test consists of repeated computations of the operation `a(i)=b(i)+q*c(i)` where `q` is constant and `i` is incremented.

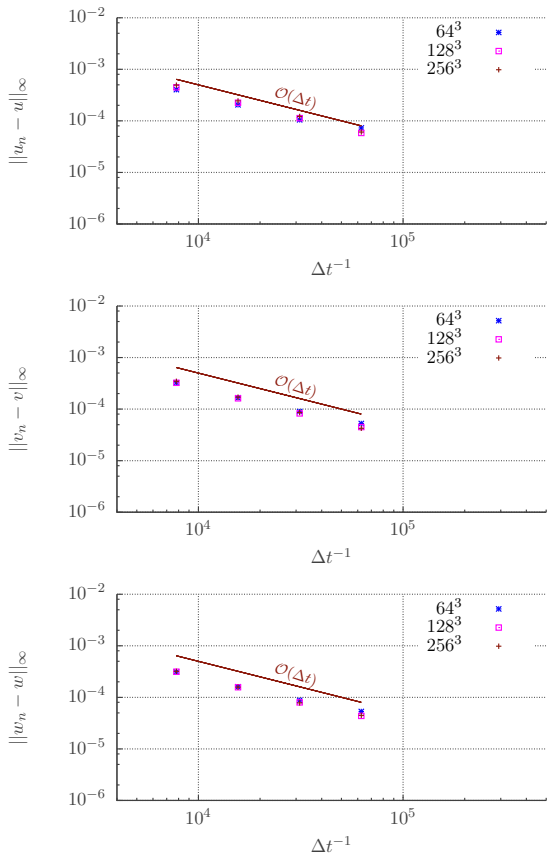



Figure 3: Time step and grid refinement on 32 processes. Top to bottom: $u$, $v$ and $w$ velocities. The maximum error of the solution, e.g. $||u_n - u||_\infty$ is plotted against the inverse of the time step. Here $u_n$ denotes the numerical solution at time step $n$ while $u$ denotes the exact solution at this time.

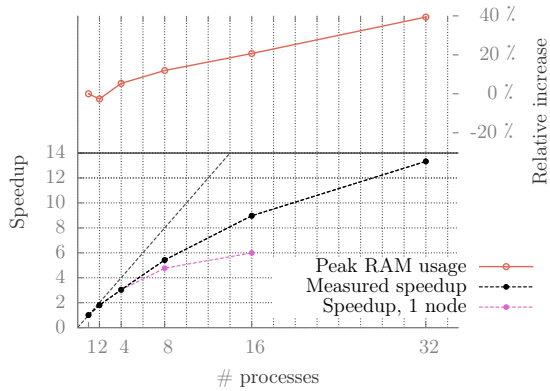



Figure 4: Strong scaling: with the left-hand-side $y$-axis, measured (black and magenta) and the optimal (dotted gray) speedup plotted against the number of processes. With the right-hand-side $y$-axis, increase in memory use.

To investigate the hypothesis that using only 2 processes per node and several nodes is better than using many processes on one node, we also tried running with 8 and 16 processes on one node. These results are plotted in magenta in Fig. 4 and confirm the hypothesis. We can conclude that even on this particular cluster with a slow (by HPC standards) interconnect, the added memory bandwidth afforded by using more nodes (thus more sockets) outweighs the penalty of increased communication between nodes. This also indicates that the results for 2 processes per node are bound by the interconnect speed, such that the speedup would be closer to the optimal (linear) scaling when run on a more tightly-coupled cluster.

### Weak scaling

The weak scaling of the code was also studied. The base case was the same manufactured solution on a $(0.5\text{ m})^3$ domain resolved with a $64^3$ grid, run on one process. Then a $(0.5\text{ m})^2\times(1.0\text{ m})$ domain with a $64^2\times128$ grid was solved with two processes, a $(0.5\text{ m})\times(1.0\text{ m})^2$ domain with a $64\times128^2$ grid was solved on 4 processes, etc. In this way, the number of grid points and the number of processes are both increased proportionally. The equations were solved for 50 time steps, and the results are shown in Fig. 5.

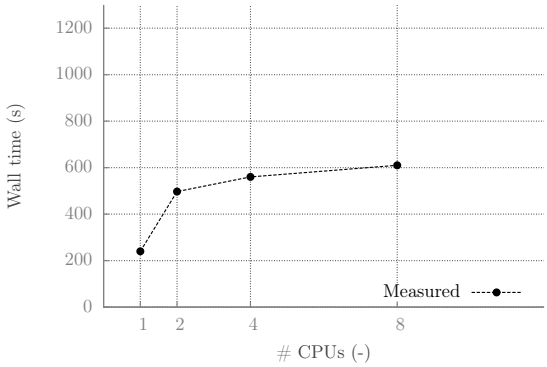



Figure 5: The weak scaling of the code as the number of processes and the number of grid points are both increased proportionally.

As is seen in this figure, there is obviously a performance hit initially; the perfect behaviour would be a flat line. This is as expected. When going from 1 to 2 processes, we go from no communication to overhead from communication. Furthermore, when going from 2 to 4 processes, there is the added overhead of intra-node communication, as opposed to the case with 2 processes where the communication is not over the network but over the CPU bus. The weak scaling seen here is quite decent. One should also be aware that it is more difficult to ensure that cases are "equivalently hard" for weak scaling than for strong scaling (Aagaard *et al.*, 2013).

### Two-phase results

As an initial test of the two-phase capabilities of the parallelized code, the CSF method was employed to simulate a 2 cm diameter drop with properties $\rho_1 = 2\text{ kg/m}^3$, $\mu_1 = 0.01$ Pa s falling through a bulk fluid with properties $\rho_2 = 1\text{ kg/m}^3$, $\mu_2 = 0.05$ Pa s. The interfacial tension was set to $\sigma = 0.01\text{ N/m}$. The domain was $(10\text{ cm})^3$ resolved by a $(128)^3$ grid, the simulation was run on 8 processes for 33900 time steps up to $t = 0.01$ s. The drop has not yet achieved a substantial falling velocity, so the spurious currents are quite visible. The result is shown in Fig. 6, where the drop is shown with the pressure superimposed on the surface, streamlines indicating the flow. A plane is shown intersecting the centre of the drop, on this plane the pressure field, velocity field and level-set function contour lines are shown. A reference vector is shown on the right.

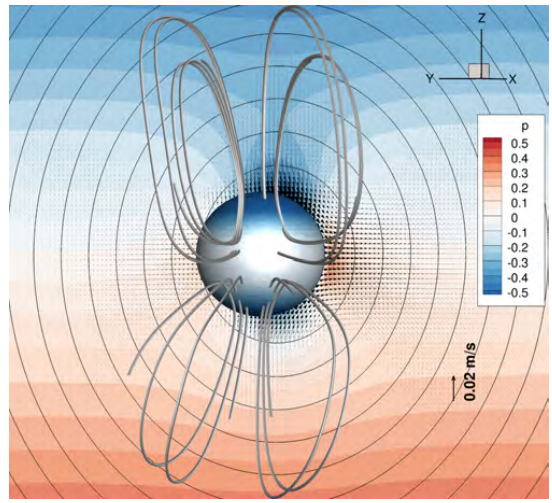



Figure 6: The falling drop after a short time (0.01 s). The pressure field is shown superimposed on the surface, and on the plane behind the drop. On this plane the velocity field and the level-set isocontours are also shown. Streamlines indicate the velocity field.

Spurious currents is a well-known challenge with the CSF method, and experience with the 2D serial code has led us to prefer the ghost-fluid method (GFM) (Kang *et al.*, 2000), which is somewhat more complicated to implement. This was not done within the scope of this paper. Nevertheless, this demonstrates that the parallel code is capable of two-phase fluid simulations with both density- and viscosity-jumps.

## CONCLUSIONS

In this paper we have discussed the parallelization of an existing serial 3D incompressible Navier-Stokes solver for two-phase flow. The PETSc DMDA domain decomposition framework has been leveraged to apply MPI parallelism, enabling the code to make use of modern HPC facilites. We have discussed the alterations that were necessary for the serial code and established a framework where these were as few as possible.

Based on this code, we have reported the strong and weak scalings for a manufactured-solution case on a cluster with dual-socket nodes and 1 Gb/s Ethernet interconnect. It is seen that the code scales rather

well, but that one should take care to maximize the number of sockets used, since the Poisson solver is bound by memory bandwidth. If this code is run on a cluster simultaneously with CPU-bound parallell codes (e.g. using Monte Carlo methods), sensible resource allocation would benefit from taking the available memory bandwidth into account. Then it would not be optimal to allocate all cores on $N$ nodes to this code, but rather e.g. 50% of the cores on $2N$ nodes, while a CPU-bound code could effectively utilize the remaining 50% of the cores.

The speedup seen in the strong scaling test (13x faster on 32 processes) is sub-linear but does not level-off. Together with the possibility of running on more tightly-coupled clusters where the behaviour should be closer to linear, and using more than 32 cores, this will give a substantial speedup and reduce the runtimes of weeks and months for the serial code to something more managable, i.e. a few days or less.

Initial tests demonstrate that the code is able simulate two-phase flow, but the ghost-fluid method (GFM) should be used instead of the CSF method currently employed, in order to minimize the spurious currents.

This effort has left us with a code that scales quite well and a framework where the remaining multi-physics components can easily be introduced. In the end this will enable future simulations of full 3D cases relevant for the fundamental understanding of electrocoalescence.

## ACKNOWLEDGEMENT

We would like to thank Matthew Knepley, Barry Smith and Jed Brown of the PETSc project for useful discussions around the framework employed here.

This work was funded by the project *Fundamental understanding of electrocoalescence in heavy crude oils* coordinated by SINTEF Energy Research. The authors acknowledge the support from the Petromaks programme of the Research Council of Norway (206976), Petrobras, Statoil and Wärtsilä Oil & Gas Systems.

## REFERENCES

AAGAARD, B.T. *et al.* (2013). "A domain decomposition approach to implementing fault slip in finite-element models of quasi-static and dynamic crustal deformation". *J. Geophys. Res.: Solid Earth*, **118(6)**, 3059–3079.

BALAY, S. *et al.* (2014). "PETSc Web page". `http://www.mcs.anl.gov/petsc`.

BRACKBILL, J. *et al.* (1992). "A continuum method for modeling surface tension". *J. Comput. Phys.*, **100(2)**, 335–354.

CHORIN, A.J. (1968). "Numerical solution of the Navier-Stokes equations". *Math. Comput.*, **22(104)**, 745–762.

DUFFY, A. *et al.* (2002). "An improved variable density pressure projection solver for adaptive meshes". Unpublished. See `http://www.math.fsu.edu/~sussman/MGAMR.pdf`.

ERVIK, Å. *et al.* (2014). "A robust method for calculating interface curvature and normal vectors using an extracted local level set". *J. Comput. Phys.*, **257**, 259–277.

GOTTLIEB, S. *et al.* (2009). "High order strong stability preserving time discretizations". *J. Sci. Comput.*, **38(3)**, 251–289.

GUERMOND, J. *et al.* (2006). "An overview of projection methods for incompressible flows". *Comp. Meth. Appl. Mech. Eng.*, **195(44)**, 6011–6045.

HENSON, V.E. and YANG, U.M. (2000). "BoomerAMG: a parallel algebraic multigrid solver and preconditioner". *Appl. Numer. Math.*, **41**, 155–177.

*hypre* (2014). *High Performance Preconditioners*. Lawrence Livermore National Laboratory. `http://www.llnl.gov/CASC/hypre/`.

JOHN, V. *et al.* (2006). "A comparison of time-discretization/linearization approaches for the incompressible Navier–Stokes equations". *Comput. Meth. Appl. Mech. Eng.*, **195(44)**, 5995–6010.

KANG, M. *et al.* (2000). "A boundary condition capturing method for multiphase incompressible flow". *J. Sci. Comput.*, **15(3)**, 323–360.

LIU, X.D. *et al.* (1994). "Weighted essentially non-oscillatory schemes". *J. Comput. Phys.*, **115(1)**, 200–212.

MCCALPIN, J.D. (1995). "Memory bandwidth and machine balance in current high performance computers". *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, 19–25.

MCCALPIN, J.D. (2014). "STREAM web site". `http://www.cs.virginia.edu/stream/`.

MOORE, G. (1965). "Cramming more components onto integrated circuits". *Electronics*, **38(8)**.

NETHERCOTE, N. and SEWARD, J. (2007). "Valgrind: a framework for heavyweight dynamic binary instrumentation". *ACM Sigplan Notices*, vol. 42, 89–100.

NETHERCOTE, N. *et al.* (2014). "Valgrind Web page". `http://valgrind.org/`.

OSHER, S. and FEDKIW, R.P. (2001). "Level set methods: An overview and some recent results". *J. Comput. Phys.*, **169(2)**, 463 – 502.

OSHER, S. and FEDKIW, R.P. (2003). *Level Set Methods and Dynamic Implicit Surfaces*. Springer, Berlin.

OSHER, S. and SETHIAN, J.A. (1988). "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations". *J. Comput. Phys.*, **79(1)**, 12 – 49.

ROACHE, P.J. (2002). "Code verification™ by the method of manufactured solutions". *ASME J. Fluids Eng.*, **124(1)**, 4–10.

ROSSINELLI, D. *et al.* (2013). "11 PFLOP/s simulations of cloud cavitation collapse". *SC*, 3.

TEIGEN, K.E. and MUNKEJORD, S.T. (2010). "Influence of surfactant on drop deformation in an electric field". *Phys. Fluids*, **22(11)**. Article 112104.

VAN DER VORST, H. (1992). "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems". *SIAM J. Sci. Comput.*, **13(2)**, 631–644.

ZHUANG, Y. and SUN, X.H. (2001). "A high-order fast direct solver for singular Poisson equations". *J. Comput. Phys.*, **171(1)**, 79 – 94.

# Paper 3

Is not included due to copyright

# Paper 4

# Influence of surfactants on the electrohydrodynamic stretching of water drops in oil

Åsmund Ervik,[1,*] Torstein Eidsnes Penne,[2] Svein Magne Hellesø,[3] Svend Tollak Munkejord,[3] and Bernhard Müller[1]

[1] *Department of Energy- and Process Engineering, Norwegian University of Science and Technology (NTNU), 7491 Trondheim, Norway*

[2] *Department of Physics, Norwegian University of Science and Technology (NTNU), 7491 Trondheim, Norway*

[3] *SINTEF Energy Research, P.O. Box 4761 Sluppen, 7465 Trondheim, Norway*

(Dated: 24th January 2016)

## Abstract

In this paper we present experimental and numerical studies of the electrohydrodynamic stretching of a sub-millimetre-sized salt water drop, immersed in oil with added non-ionic surfactant, and subjected to a suddenly applied electric field of magnitude approaching 1 kV/mm. By varying the drop size, electric field strength and surfactant concentration we cover the whole range of electric capillary numbers ($Ca_E$) from 0 up to the limit of drop disintegration. The results are compared with the analytical result by G.I. Taylor (*Proc. R. Soc. A* **280**, 383 (1964)) which predicts the asymptotic deformation as a function of $Ca_E$. We find that the addition of surfactant damps the transient oscillations and that the drops may be stretched slightly beyond the stability limit found by Taylor. We proceed to study the damping of the oscillations, and show that increasing the surfactant concentration has a dual effect of first increasing the damping at low concentrations, and then increasing the asymptotic deformation at higher concentrations. We explain this by comparing the Marangoni forces and the interfacial tension as the drops deform. Finally, we have observed in the experiments a significant hysteresis effect when drops in oil with large concentration of surfactant are subjected to repeated deformations with increasing electric field strengths. This effect is not attributable to the flow nor the interfacial surfactant transport.

PACS numbers: 47.11.-j,47.15.G-,47.55.D-,47.55.dk,47.65.-d

* asmunder@pvv.org

## I. INTRODUCTION

Surfactants are ubiquitous in two-phase fluid flows. Take for instance a single drop falling through a viscous fluid, perhaps the simplest and most widely studied two-phase flow configuration. While the classic results by Hadamard [1] and Rybzynski [2] give the analytical result for the flow field in this case, experimental investigations mostly fail to agree with this result. The discrepancy is attributed to trace surface-active contaminants, found even in the most purified of liquids. It is natural, then, also to consider the effects of surfactants on the more complicated case of electrohydrodynamic deformation of a conducting drop falling in an insulating oil.

The case of a drop deforming in an electric field is interesting, not only as an intriguing physical phenomenon of which our understanding can be improved, but also for applications e.g. to chemical processing equipment such as electrocoalescers [3–5]. A deeper understanding of the physical processes at play in this system could lead to improved coalescer equipment and reduced emissions.

We will consider here experiments and simulations of sub-millimetre-sized drops of brine falling in a highly refined oil with added surfactant, studying the drop deformations and oscillations induced by square voltage pulses of varying amplitude applied to parallel electrodes above and below such a drop.

When performing these studies of drop deformations, it is crucial to have a system which is well characterised in terms of the fluid and the interfacial properties. To overcome the uncertainties associated with unknown trace contaminants acting as surface-active agents, we deliberately add a non-ionic surfactant (Span 80) in known, small quantities. The interfacial tension as a function of surfactant concentration is then measured, together with the bulk properties, to give a well-characterised system.

There is a large amount of research on the deformations of drops in electric fields, using analytical, experimental and numerical techniques; we will not summarise all of it here. The review by Melcher and Taylor [6] covers the fundamentals in a thorough fashion, while the review by Saville [7] gives an update with more recent results in the field. However, when surfactants are added to this picture, the literature is not so extensive. Previous authors [8, 9] have investigated the influence of surfactants on the electrohydrodynamic stretching experimentally, but they have been limited to considerations of the static (equilibrium)

deformation, as well as drop sizes above 1 mm in diameter, and a limited number of observations. Computational studies in the literature, namely previous work by Teigen *et al.* [10], and the paper by Nganguia *et al.* [11] which finds good agreement with [10], have also been focused on the static deformation. Note that the numerical code used in this paper is the same as in [10].

Taking a step further, we consider here also the dynamical behaviour of the stretching drops, in particular the effects of the surfactant concentration on the damping of the drop oscillations. We work with drops smaller than 1 mm in diameter. We report results for many drop deformations, almost 300 for the experiments and 44 representative cases for the simulations.

This work is an extension of our initial investigation [12], where five cases of the electrohydrodynamic deformation of drops in insulating oil were studied. In the present work we have extended this analysis to a parameter study of the factors influencing the deformation and the deviations from the classical result by Taylor [13], which does not take surfactants into account. The analytical result by Taylor has been found to agree very well with subsequent results, see e.g. [14], and for this reason we use it as a supporting line in the plots and analysis throughout the paper. Following Taylor, we use the dimensionless electric field strength $\zeta = \sqrt{Ca_E}$ in the following.

The results presented here show that the deviation from Taylor's expression is negligible below dimensionless electric field strengths of $\zeta \approx 0.4$, while above this threshold they become significant. We demonstrate that drops in the presence of surfactants may be deformed beyond the stability limit given by the Taylor theory. Finally we study the effect of the surfactant concentration, and the effects of Marangoni stresses on the damping of drop oscillations. Our results indicate that small concentrations of surfactant give a significant increase in the damping whilst having but a small effect on the equilibrium (static) shape. Also, for the highest surfactant concentration used here, we observe in the experiments a significant hysteresis effect of repeated stretchings. This effect is not seen in the simulations, so it cannot be explained by the hydrodynamics and the surfactant transport processes which are modelled by our approach.

## II. THEORY

The flow of single-phase oil or water can be described by the incompressible Navier-Stokes equations

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{u} = -\frac{\boldsymbol{\nabla} p}{\rho} + \frac{\eta}{\rho}\boldsymbol{\nabla}^2 \boldsymbol{u} + \boldsymbol{f}, \tag{2}$$

where $\boldsymbol{u}$ is the velocity field, $p$ is the pressure, $\rho$ is the density, $\eta$ is the dynamic viscosity, and $\boldsymbol{f}$ is the acceleration caused by some body force, e.g. the gravitational acceleration. This description can be extended to a two-phase flow by incorporating three things, namely that there is an interface separating the two fluids, that the fluids may have different viscosities $\eta_1, \eta_2$ and densities $\rho_1, \rho_2$, and finally the effects of interfacial tension and interfacial tension gradients. We mark the drop properties with subscript $_1$ and the bulk properties with $_2$, and denote the interfacial tension by $\gamma$. The viscosity difference and the interfacial tension $\gamma$ contribute to jumps across the interface in various properties such as the pressure; this is detailed in equations (16) to (18) below. Mathematically, this can be incorporated into the Navier-Stokes equations as a singular contribution to $\boldsymbol{f}$ in equation (2).

This system admits two dimensionless groups, which we may take to be the Reynolds number $Re$ and the Ohnesorge number $Oh$. The Reynolds number is of interest for a falling drop, where it is defined as $Re_{\mathrm{D}} = \rho_2 u_T D/\eta_2$, $u_T$ being the terminal velocity and $D$ being the drop diameter. For the drops considered here, the Reynolds number is small ($Re_{\mathrm{D}} < 1$), meaning that the inertial term in equation (2) is unimportant for the flow at terminal velocity.

For an oscillating drop, the Ohnesorge number is an important quantity; some authors use the inverse of the Ohnesorge number as the "oscillation Reynolds number" $Re_{\mathrm{osc}}$. We use the definition $Oh = \eta_2/\sqrt{\rho_2 \gamma D}$, since the ambient fluid is much more viscous for the cases considered here. For the oscillations, the Ohnesorge number is also small ($Oh < 0.2$), but here the inertial term is important since small $Oh$ corresponds to large $Re_{\mathrm{osc}}$.

When considering a single small (i.e. spherical) drop falling in a clean fluid at low Reynolds number, the terminal velocity as well as the flow in the entire domain is given analytically by the results that Hadamard [1] and Rybzynski [2] obtained independently.

Their result for the terminal velocity, assuming an unbounded domain, is

$$v_{\mathrm{T,HR}} = \frac{(\rho_1 - \rho_2)\boldsymbol{g}D^2(\eta_1 + \eta_2)}{6\eta_2(3\eta_1 + 2\eta_2)}. \tag{3}$$

Experimental results for the terminal velocity, however, tend to not agree with this result [see e.g. 15, Fig. 1], but a closer agreement is found with the formula derived by Stokes [16] for a hard sphere falling in an unbounded domain,

$$v_{\mathrm{T,S}} = \frac{(\rho_1 - \rho_2)\boldsymbol{g}D^2}{18\eta_2}. \tag{4}$$

Indeed Hadamard himself was aware of this discrepancy, as is evident when reading the closing paragraph of his 1911 paper, wherein he mentions poor agreement with as-yet unpublished experiments he had performed.

We note that for $\eta_1 < \infty$, the graphs of $v_{\mathrm{T}}(D)$ given by equations (3) and (4) only intersect at $D = 0$, and thus the terminal velocity of a falling drop is an observable quantity that can determine if a system is clean or not. An experimental observation closer to equation (4) indicates a contaminated system, which is indeed the observation for most fluid combinations. It is noteworthy that the experiments which have obtained values agreeing with equation (3) are for quite singular fluid combinations, e.g. mercury drops in glycerine [17]. Experiments reported in our previous paper [12] indicate that the pharmaceutical grade white-oil used here, Marcol 52, contains contaminants that make small water drops fall like hard spheres.

The currently accepted explanation [see 18, pp. 35-41] of this phenomenon is that trace contaminants in the system act as surfactants which are swept along the interface by the flow, creating an interfacial-tension gradient which results in a Marangoni force, with the end result that the drop interface is immobile. Since the nature of these trace contaminants are not known, we deliberately add to the oil a known amount of a non-ionic surfactant, Span 80, such that we obtain a well-described fluid system.

The interfacial tension, $\gamma$, can be related to the bulk concentration of surfactant, $\Lambda$, using the Szyszkowski [19] equation of state (EoS):

$$\gamma(\Lambda) = \gamma_0 \left[1 - \beta \ln\left(1 + \frac{\Lambda}{a_L}\right)\right], \tag{5}$$

where $\gamma_0$ is the interfacial tension without surfactants, $\beta = R_{\mathrm{gas}}T\Gamma_\infty/\gamma_0$ is the interfacial elasticity, and $a_L = k_{\mathrm{des}}/k_{\mathrm{ads}}$ is the ratio between the adsorption and desorption coefficients of

the surfactant. In the expression for $\beta$, $\Gamma_\infty$ is the maximum possible interfacial concentration of surfactant, $R_{\text{gas}}$ is the universal gas constant, and $T$ is the temperature (in Kelvin). The parameters $\beta, a_L$ of this EoS may be computed by fitting to experimental data; note that this also determines $\Gamma_\infty$ when the temperature is known.

The equilibrium interfacial concentration can subsequently be calculated as

$$\Gamma = \Gamma_\infty \frac{\Lambda}{\Lambda + a_L}. \tag{6}$$

The relationship between interfacial concentration and interfacial tension is then given by the Langmuir EoS:

$$\gamma(\Gamma) = \gamma_0 \left[ 1 + \beta \ln \left( 1 - \frac{\Gamma}{\Gamma_\infty} \right) \right]. \tag{7}$$

For a detailed review of these equations and their derivation, see e.g. [20, pp. 47–50]. In figure 2 in the next section we plot this equation, with parameters obtained by fitting equation (5) to the experimental data of interfacial tension as a function of concentration, together with the experimental data. It is seen that the fit is very good.

In the present case we consider the surfactant to be insoluble, and we restrict ourselves both in simulations and experiments to surfactant concentrations which are below the critical micelle concentration (0.02 wt% for our system). An insoluble surfactant is a good approximation when the time scales for adsorption-desorption are long when compared to the deformation time scales [21, 22]. This is the case here, since the time it takes to reach equilibrium for the measurements of interfacial tension is in the order of minutes, while the time period of the drop deformations discussed is on the order of milliseconds. We denote the non-equilibrium interfacial surfactant concentration by $\xi$. The initial value of $\xi$ is given by equation (6), and the concentration profile $\xi(\boldsymbol{x})$ evolves according to an advection-diffusion equation which is restricted to the interface (see e.g. [23]), namely

$$\frac{\partial \xi}{\partial t} + u_i \frac{\partial \xi}{\partial x_i} - n_i n_j \frac{\partial u_j}{\partial x_i} \xi = D_\xi \left( \frac{\partial^2 \xi}{\partial x_i \partial x_i} - \frac{n_i n_j \partial^2 \xi}{\partial x_i \partial x_j} + \kappa \, n_i \frac{\partial \xi}{\partial x_i} \right), \tag{8}$$

where we employ the Einstein summation convention. $u_i$ and $n_i$ denotes the components of the velocity $\boldsymbol{u}$ and the normal vector $\boldsymbol{n}$, respectively. $\kappa = \boldsymbol{\nabla} \cdot \boldsymbol{n}$ is the interfacial curvature. $D_\xi$ is the surfactant interfacial diffusion coefficient, a parameter which is very difficult to measure. Fortunately the solutions of this equation are quite insensitive to the value of this constant as long as it is small. Here we use the value $5 \cdot 10^{-7}$ m$^2$/s, which is of the same magnitude as that reported e.g. by Sakata and Berg [24] (albeit for different surfactants).

For an oil-soluble surfactant with a long aliphatic tail such as Span 80, it makes sense that the diffusion coefficient is quite low when the oil is quite viscous.

With this approach, the Gibbs elasticity is taken into account, and its magnitude can be computed as $\beta\gamma$ [22]. This only takes into account the elasticity caused by the change in interfacial tension given by a change in the drop area. Other physical mechanisms, such as reorientation of surfactant molecules at the interface, can lead to additional effects, and even cause a phase transition in the surfactant layer [25]. These mechanisms are not accounted for in the model used here.

As is the case with the falling drop, the oscillating fluid drop has received much attention in the scientific literature. The earliest work is due to Bidone [26] who experimentally studied the breakup of thin liquid jets and the oscillating drops formed by these. Parts of that paper are reproduced in the seminal study by Rayleigh [27] who presented a theoretical analysis of why thin liquid jets break up, and that instability now bears Rayleigh's name. In that paper he also investigated the oscillations of the liquid drops produced after jet breakup, and derived an expression for the frequency of an oscillating drop valid at small deformations, zero viscosity and no outer fluid.

The analysis was later extended by Lamb [28] to include the damping of oscillations for a weakly viscous drop, and then by Chandrasekhar [29] to any viscosity, but still only for a liquid drop in vacuum or a dilute gas. The extension to a liquid drop immersed in another liquid was then done by Miller and Scriven [30], and extended again by Prosperetti [31] to a result valid at all times, not just asymptotically. We note that this analysis has been applied for measuring the interfacial tension and viscosity of two-phase fluid systems, a technique first described by Ronay [32]. In all these descriptions, the surface is decomposed into spherical harmonics, and they differ essentially only in how the coefficients of these harmonics are distributed and how they evolve in time. Another common feature is that all considerations of damping lead to the result that the fundamental mode of stretching (i.e. an elongated drop) is damped most slowly, and that the higher modes have significantly lower amplitude and faster damping. It is notable that experiments by Apfel *et al.* [33] which show more exotic oscillations, including e.g. an hour-glass shape and a dimpled sphere, had to be performed in a micro-gravity environment. A third common feature is that they all work in the limit $Oh \to 0$, i.e. they assume that the viscosity is small, and so they linearise the Navier-Stokes equations.

While this type of analysis is a popular way of attacking the problem, it is unfortunately not applicable in the present case. This is because we are dealing with a very viscous outer fluid, so the oscillations are damped out in one or two periods. It is well-known that the theory discussed in the previous paragraph is not applicable in the strict sense to the initial value problem, since the initial condition is not in the solution space. Nevertheless, as demonstrated by Prosperetti [31], applying the linearised theory starting from the end of the first period of oscillation gives good results. Unfortunately, after the first period, the oscillations we consider here are all but extinguished. We thus find ourselves in the situation where this theory cannot assist our understanding of the problem at hand. See also figure 13 in the appendix, which shows that the oscillation frequency observed in the simulations here is significantly lower than that obtained with the formula in [28, p. 475].

Another seminal approach from a different point of view is the study by Taylor [13] who did a theoretical analysis of the electrohydrodynamic stretching of a clean conducting drop in a perfect dielectric medium. His result predicts the asymptotic drop deformation as a function of the electric field strength, radius, permittivity of the oil and interfacial tension, all combined into a dimensionless electric field strength $\zeta$. This is equivalent to the square root of the electric capillary number, $\zeta = \sqrt{Ca_E}$, where $\zeta$ is defined as

$$\zeta = \bar{E}\sqrt{\epsilon\epsilon_0 D/\gamma}, \tag{9}$$

and $\bar{E}$ is the uniform electric field that is present far away from the drop. Note that papers from that era work in electrostatic units, where the numerical value of $\epsilon_0$ is 1, so it is frequently omitted from their formulae. Note also that some authors use the drop radius rather than the diameter here.

We may compare $\zeta^2$ to the capillary number computed from the terminal velocity, $Ca = \eta_2 u_T/\gamma$, giving us an impression of the relative importance of the external flow versus the electric field as far as the drop shape is concerned. Using numbers relevant to the situation at hand, we estimate a typical value of the hydrodynamic capillary number to be $Ca \approx 0.007$, while a typical value for the electric capillary number is $\zeta^2 \approx 0.25$, indicating that the electric field has a much greater influence on the drop shape than the deformation due to the external flow. One may thus neglect the effects of the external flow when considering the drop deformations.

In his analysis of drop deformation, Taylor considered a clean drop at rest in a medium

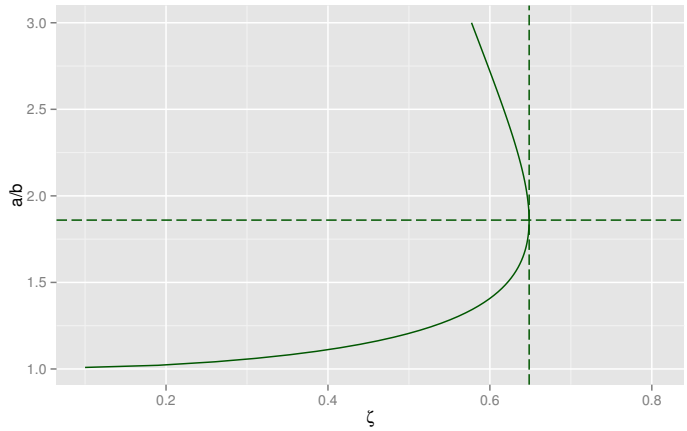



Figure 1: The static deformation predicted by Taylor's theory.

with the permittivity of free space, approximated the drop shape at equilibrium as ellipsoidal, and assumed that the value of the difference between interfacial tension and electrical stress at the interface is equal at the poles and at the equator. He then derived an implicit formula which predicts the drop elongation $a/b$ as a function of $\zeta$. In this context, $a$ and $b$ denote the semi-major and semi-minor axis of the drop, respectively, see also figure 3b in the next section.

That implicit formula may be given e.g. as the zero level of the function $f(\zeta, a/b)$,

$$f\left(\zeta, \frac{a}{b}\right) = 2\left(\frac{a}{b}\right)^{-4/3}\sqrt{2 - \left(\frac{a}{b}\right)^{-1} - \left(\frac{a}{b}\right)^{-3}}\, I - \zeta, \tag{10}$$

$$I = \frac{1}{2}e^{-3}\ln\left(\frac{1+e}{1-e}\right) - e^{-2}, \tag{11}$$

$$e = \sqrt{1 - (a/b)^{-2}}. \tag{12}$$

This predicts a limit to the static deformation at $a/b \approx 1.86$ and $\zeta \approx 0.65$; at higher applied field strengths the drop does not reach an equilibrium state, but is torn apart. Taylor showed that this limit agrees with experiments done with drops in air. The theoretical result by Taylor is shown in figure 1 together with a horizontal line at $a/b = 1.86$ and a vertical line at $\zeta = 0.6485$.

For the numerical model, including the effects of an electric field on the drop requires knowledge of this field inside the simulation domain. Even though we consider a conducting

drop in a dielectric medium, we may model the situation as two dielectric media with a very high permittivity ratio [6]. We use here a numerical value of 1000 for the relative permittivity of the conducting liquid; this value is not important as long as it is much larger than that of the dielectric liquid. The model validity is confirmed by the fact that the calculated field lines inside the drop are indeed straight, parallel lines in the direction normal to the electrodes, as seen in figure 4b.

To obtain the electric field, we may then proceed by solving a Laplace equation for the electric potential $\Psi$, with the applied voltage as boundary conditions at the top and bottom of the domain, and $\boldsymbol{\nabla}\Psi \cdot \boldsymbol{n} = 0$ at the vertical boundaries of the domain. To wit:

$$\boldsymbol{\nabla} \cdot (\epsilon\epsilon_0 \boldsymbol{\nabla}\Psi) = 0, \tag{13}$$

where $\epsilon$ is the relative permittivity and $\epsilon_0$ is the permittivity in vacuum. Note that we keep $\epsilon$ inside the divergence operator here, even though it is piecewise constant, since this is how the discontinuity is handled by the numerical method.

The Maxwell stress tensor, $\overleftrightarrow{M}$, can then be calculated from the electric field $\boldsymbol{E} = -\boldsymbol{\nabla}\Psi$. Neglecting the magnetic field, which is not of interest here, we have

$$\overleftrightarrow{M} = \epsilon\epsilon_0\Big(\boldsymbol{E}\boldsymbol{E} - \frac{1}{2}(\boldsymbol{E}\cdot\boldsymbol{E})\overleftrightarrow{\boldsymbol{I}}\Big). \tag{14}$$

This stress gives a spatially varying contribution to e.g. the jump in the pressure across the interface, which will distort a drop from its spherical shape.

All in all, the formulation presented here takes into account the effects of interfacial tension $\gamma$, the applied electric field, and the Marangoni effect that arises from an interfacial-tension gradient. The jumps across the drop interface in various properties are then given as [10, 34]

$$[\![\boldsymbol{u}]\!] = 0, \tag{15}$$

$$[\![p]\!] = 2[\![\eta]\!]\boldsymbol{n}\cdot\boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{n} + \boldsymbol{n}\cdot[\![\overleftrightarrow{M}]\!]\cdot\boldsymbol{n} - \gamma\kappa, \tag{16}$$

$$[\![\Psi]\!] = 0, \tag{17}$$

$$\begin{aligned}[\![\eta\boldsymbol{\nabla}\boldsymbol{u}]\!] = [\![\eta]\!]\Big(&(\boldsymbol{n}\cdot\boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{n})\boldsymbol{n}\boldsymbol{n} + (\boldsymbol{n}\cdot\boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{t})\boldsymbol{n}\boldsymbol{t} \\ &- (\boldsymbol{n}\cdot\boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{t})\boldsymbol{t}\boldsymbol{n} + (\boldsymbol{t}\cdot\boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{t})\boldsymbol{t}\boldsymbol{t}\Big) \\ &- (\boldsymbol{t}\cdot\boldsymbol{\nabla}_\iota\gamma)\boldsymbol{t}\boldsymbol{n},\end{aligned} \tag{18}$$

In these expressions, $\boldsymbol{n}$ and $\boldsymbol{t}$ are the normal and tangent unit vectors to the interface. Expressions such as $\boldsymbol{\nabla u}$ and $\boldsymbol{nn}$ denote rank-two tensors formed by the outer product, so e.g. $\boldsymbol{\nabla u} \cdot \boldsymbol{n}$ denotes such a tensor acting on a vector. We use the convention that a normal vector on a drop points towards the external fluid, and that the jump $[\![-]\!]$ is the difference between the external and the internal properties, e.g. $[\![\eta]\!] = \eta_2 - \eta_1$. Note that these expressions have been written in a form which results in faster computer code, see [35, p. 14] for details. The interface is denoted by $\iota$ here, so $\boldsymbol{\nabla}_\iota$ is the gradient along the interface. For the sake of completeness, we mention that an additional term $-(\boldsymbol{t} \cdot [\![\overleftrightarrow{M}]\!] \cdot \boldsymbol{n})$ contributes to equation (18) when the fluids are not perfect dielectrics, but rather leaky dielectrics. This is considered e.g. in [36, 37], and gives an electric contribution to the tangential force at the interface.

## III. METHODS

### A. Experimental methods

Experiments were performed with brine drops (3.36 wt% NaCl added to Milli-Q purified water) immersed in Marcol 52 oil (ExxonMobil), which is a purified and hydrogenated hydrocarbon oil with very low content of surface-active components. Span 80 non-ionic surfactant (Sigma-Aldrich) was added. The densities were measured with an Anton Paar DMA 5000 density meter. The viscosity of the oil was measured with an Anton Paar MCR 102 rheometer. Tabulated values from White [38] were used for brine viscosity. Experiments were done with temperature control at 21.5°C, where viscosity and density of water were 1.03 mPa·s and 1023.6 kg/m$^3$, respectively, and those of the oil were 12.4 mPa·s and 832.3 kg/m$^3$, respectively. The relative permittivity of Marcol 52 was taken to be 2.13, as per the data sheet supplied by the manufacturer.

Interfacial tension was measured with a SIGMA 703D tensiometer with a DuNuoy ring, for different Span 80-concentrations, with selected values shown in table I. All data points are given in the supplementary information. Here wt% means weight percent. From these measurements, the critical micelle concentration was determined to be 0.020 wt%, and we limit ourselves to concentrations below this value.

$\beta$ and $a_L$ in equation (5) were determined by fitting this equation to the experimental

| wt% Span 80 | Interfacial tension [mN/m] |
|---|---|
| 0.030 | 10.0 |
| 0.020 | 10.1 |
| 0.015 | 13.9 |
| 0.010 | 18.8 |
| 0.001 | 29.4 |

Table I: Interfacial tension between water and oil for different surfactant concentrations

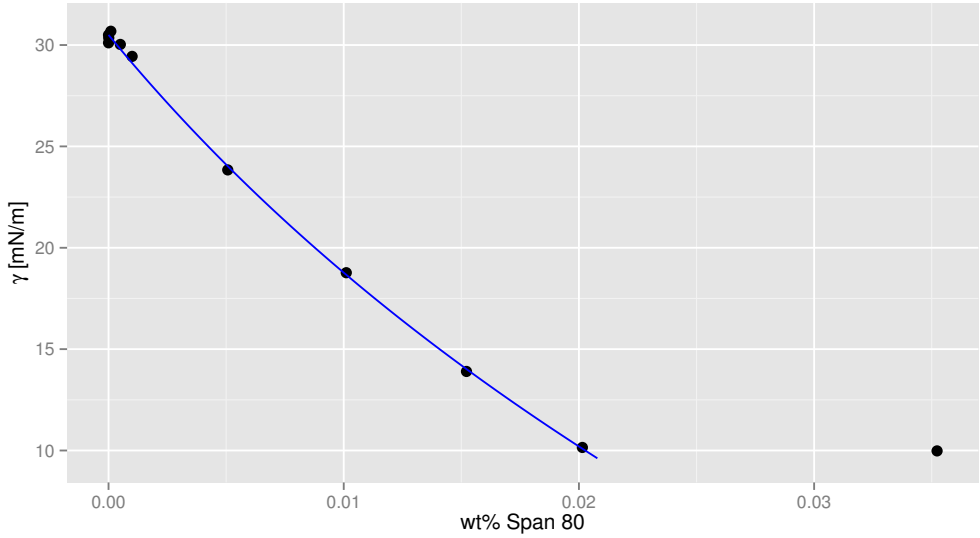



Figure 2: Experimental measurements of interfacial tension (points) and the Langmuir EoS (7) fitted to these (line). Note that the interfacial tension is constant above 0.02 wt%, indicating that this is the CMC.

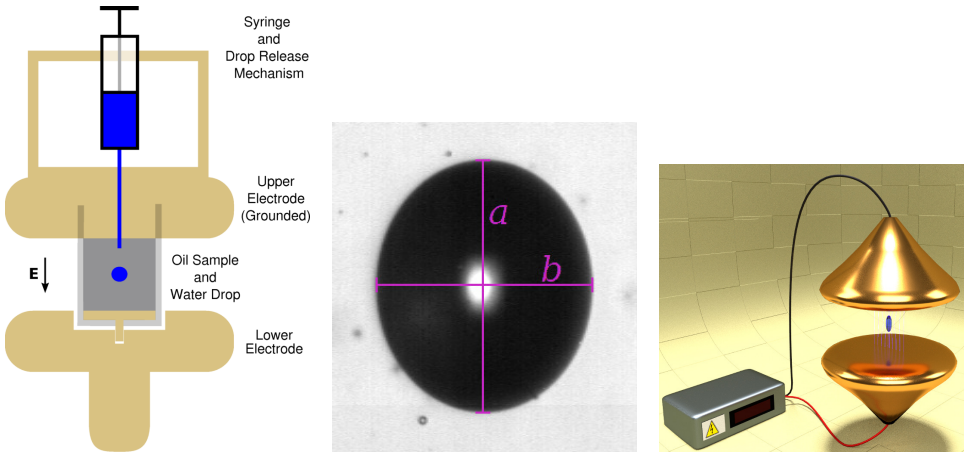
measurements using non-linear least-squares. See the plot of the data points and the fitted equation in figure 2; note in particular that the interfacial tension is constant above 0.02 wt%, confirming that this is the critical micelle concentration (CMC). It is somewhat difficult to tell whether the point at 0.02 wt% is a little above or a little below the CMC; thus we have tested the sensitivity of the curve fit to this point by also computing a fit with this

point omitted. At the highest concentration studied here, 0.016 wt%, this change in the curve fit produced a change in the interfacial tension predicted by the EoS of 0.1 mN/m, i.e. less than 1% and within the experimental uncertainty. Accordincly, the small uncertainty about the exact value of the CMC has no influence on the results presented in this paper.

In addition to fitting the EoS, the interfacial area available to each surfactant molecule at the critical micelle concentration, $A_{\mathrm{CMC}}$, was estimated from the slope of the Gibbs isotherm as it approaches the CMC [39]. For further details, see the supplementary information which contains the script used for fitting and the experimental data. When comparing with the results by Peltonen and Yliruusi [40], we find good agreement for the values of $A_{\mathrm{CMC}}$ and the critical micelle concentration (CMC) obtained here, 30.5 Å$^2$ and 0.020 wt%, respectively.

The deformation of the water drops was observed as they fell in the 15 mm gap between an upper and a lower horizontal metal electrode. Drops were produced from a screw-in syringe connected to a glass capillary tube made hydrophobic by a silane coating; this tube protruded through a small hole in the upper electrode. A series of square voltages with different amplitudes was applied to the lower electrode, creating an electric field $\boldsymbol{E}$ that distorted the drop. The voltage pulse shape was generated in MATLAB and sent over a serial connection to a Stanford Research DS340 signal generator connected to a TREK 2020B high voltage amplifier. The voltage pulse shapes were either rising or falling, and included both positive and negative pulses; i.e. the pulses were bipolar. The length of each pulse was 25 ms, with a 25 ms pause between pulses. The amplitudes were defined as fractions of a maximum amplitude $V_0$, e.g. $V_0 = 10$ kV and fractions 2/4, 3/4 and 4/4 times $V_0$, giving pulses that look like  or  . Typically 6 different amplitudes (fractions) were used. This application of different voltages resulting in different stretchings of the same drop is the only practical way of studying the effect of varying the electric field strength at constant drop radius. It also allows us to study possible hysteresis effects of the stretching on the surfactant on the interface. Such effects have been reported previously, e.g. in [40] for the Span 80 surfactant used here.

When applying several voltage pulses to the electrodes it is desirable to keep the drop in the camera field-of-view for as long as possible. To achieve this, a moving stage setup was used, comprised of a Newport (M-)IMS-V linear stage to move the test cell containing the fluid system upwards a constant velocity, and a Newport XPS Series Motion Controller to manually match the velocity of the moving stage to the terminal velocity of the drop.




(a) Experimental setup. (b) Captured image of drop. (c) 3D rendering of setup.

Figure 3: Schematic showing a side view of the experimental setup, an example of a captured drop image shown with the major and minor axes $a$ and $b$ superimposed, and a 3D rendering with a simulated droplet and electric field shown between the electrodes. The drop size is exaggerated in both (a) and (c).

A side view of the experimental setup is shown in figure 3a, and a 3D rendering is shown in figure 3c. To avoid unnecessary clutter, the temperature control bath, optical setup and the linear stage are omitted in both of these figures, and in figure 3c the cuvette containing oil and the syringe mechanism are also omitted. The drop size relative to the setup is exaggerated in both figures.

To record high-speed movies of a falling drop, a Cheetah CL near-infrared camera was used with an Infinity KS2 long-range microscope lens, and a collimated light source was placed on the opposite side of the cuvette. The camera recorded a frame of $640 \times 512$ pixels at 1730 frames per second. The high-speed movies were recorded in the Streams 7 software, together with the voltage pulse from the signal generator and the velocity and position of the moving stage. These electrical signals were captured using a National Instruments PCI-6052E DAQ board.

To determine the drop deformations from the high-speed images, the Spotlight image analysis software was used. With this software, the dimensions of the major axis $a$ and minor axis $b$ of a deformed drop can be determined, as seen in figure 3b, where the axes

are superimposed on an image of an elongated drop. This then gives the ratio $a/b$ as a measure of the deformation; note that this measure does not make any assumptions about the drop shape. See also the supplemental material in movie 1 which shows a video of a drop deformation cycle together with an animated plot of $a/b$ as a function of time.

The various uncertainties that affected the measurement of $a/b$ were analysed in a fashion similar to that used in [41] and Gaussian error propagation was then used to compute the uncertainty in $a/b$ [42]. This uncertainty was found to be independent of $a/b$, but dependent on the initial drop radius, which is sensible. The relative error in $a/b$ was largest for the smallest drops under consideration, at 3.4%, and smallest for the largest drops considered, at 2.0%.

The experimental procedure for studying the drop deformation in an electric field was comprised of the following steps:

**1.** Move the test cell using the linear stage such that the tip of the needle used for generating droplets is in the top of the camera's field-of-view. Use the screw-in plunger to create a droplet of the desired size.

**2.** Wait for 1 minute to allow the equilibration of surfactants at the drop interface. Arm the camera such that it starts recording 10 ms before the first voltage pulse is applied.

**3.** Use an electromagnet to jerk the glass needle upwards, releasing the drop.

**4.** As the drop falls through the view of the camera, adjust the upwards velocity of the moving stage to match the terminal velocity of the drop, keeping the drop in the centre of the image. The drop falls for approx. 0.5 seconds before the voltage pulse train is applied, allowing ample time for any initial oscillations to be damped away.

**5.** Trigger the voltage pulse train. The camera is also controlled by this trigger and records a movie of the drop being deformed.

**6.** Post-process the recorded movie to extract $a$, $b$ as functions of time.

A remark is in order with regards to the waiting time for equilibration of surfactants at the drop. As stated, a waiting time of one minute is used in these studies. If we are to compare this with some intrinsic time scale, we may consider $\tau_D = \Gamma^2/(\Lambda^2 D_B)$ [43], with $\Gamma$ taken at some surfactant concentration $\Lambda$, say the highest used in these experiments (0.016 wt%). The value of $D_B$, the bulk diffusion coefficient of Span 80 in Marcol 52 oil, is not known. Since the Span 80 molecule is not much larger than the alkanes in the oil, we may

use as a rough estimate the self-diffusion coefficient of the tail of the Span 80 molecule, namely oleic acid, which gives $D_B \approx 10^{-10}$ m$^2$/s [44]; of the same order of magnitude as e.g. the diffusion coefficient of $C_{12}E_6$ surfactant in water [45]. This gives a time scale of $\tau_D \approx 3 \cdot 10^5$ s, i.e. 3.5 days. It would be impractical to wait for such a long time between each drop was produced.

Fortunately, the transport of surfactants to the interface is greatly accelerated once the drop starts falling, since the velocity boundary layer decreases the length of the diffusion boundary layer. Note that the length of the diffusion boundary layer as estimated above is $l = \Gamma/\Lambda = 20\mu$m, which is comparable to the drop radius of 250–500 $\mu$m. To quantify the increase in surfactant transport to the interface, we may consider the Schmidt number, i.e. the ratio between the viscous and the molecular diffusion rates, which in this system is $Sc = 2 \cdot 10^5$. This indicates that the combination of the one minute waiting time and the subsequent falling time of 0.5 s between drop release and the start of the first deformation should be sufficient to ensure the interfacial surfactant concentration is equilibrated before the deformations commence.

### B. Simulation and numerical methods

An in-house code was used to solve the Navier-Stokes equations (1) and (2) numerically. The simulations reported here are done in axisymmetry. See Teigen *et al.* [10], Teigen and Munkejord [36, 46] for validation of the methods and implementation used here.

The equations are discretised on a structured, uniform, staggered grid using the finite-difference method. For the convective terms, the fifth-order WENO scheme [47, 48] is employed. For the other terms a standard second-order central-difference scheme is used. The pressure and velocity fields are coupled using the classical projection method due to Chorin [49], which gives a Poisson equation for the pressure. This Poisson equation is solved here using the BoomerAMG (Algebraic MultiGrid) preconditioner [50] and the BiCGStab (Bi-Conjugate Gradient Stabilised) iterative solver [51]; we use the Hypre [52, 53] and PETSc [54] libraries for these methods.

The equations are then integrated in time using an explicit Runge-Kutta method which has the strong stability preserving (SSP) property, namely SSPRK(2,2) in the terminology of Gottlieb *et al.* [55]. Although this method is second order in time, the overall scheme

is only first order in time due to the irreducible splitting error from the Chorin projection method [see e.g. 56]. The reason for still using a second-order Runge-Kutta method is the enlarged stability domain as compared to the forward Euler method. To sum up, the present method is first-order in time and second-order in space. We are not aware of any explicit second-order in time projection method for the incompressible Navier-Stokes equations. It is only very recently that a feasible implicit scheme has been developed [57] for two-phase flow simulations with the ghost-fluid method, allowing simulations with higher temporal order.

To capture the position of the interface between the two fluids, the level-set method [58, 59] is used with the high-order constrained reinitialisation method [60] and the velocity extrapolation procedure [61]. With the interface position known, the ghost-fluid method [62, 63] is used to enforce the jumps specified in equations (16) to (18) across the interface in sharp fashion.

This formulation takes into account the effects of interfacial tension $\gamma$, the applied electric field, and the Marangoni effect that arises from an interfacial tension gradient. The surfactant concentration along the interface, $\xi$, is determined by solving the advection-diffusion equation (8). The interfacial tension $\gamma$ is then determined by the surfactant concentration according to equation (7), using equation (6) with $\xi$ in place of $\Lambda$.

There exists in the literature a large number of interface tracking and interface capturing methods: front tracking [64], volume of fluid [65], level set [58], phase field [66] to name but a few. Many of these methods have been applied to the study of electrohydrodynamic deformation of conducting drops [67–70]. The combination used here of the level-set method with the sharp interface handling of the ghost-fluid method is chosen because it is easy to implement, has been extensively verified and validated in the literature (including by ourselves using the present numerical code), and because it gives an accurate handling of interfacial tension and the jumps in the density and viscosity across the interface.

## IV. RESULTS

### A. Parameter studies on drop deformation

As mentioned in the introduction, five distinct cases of drop deformation in the presence of surfactants were studied in our previous paper [12], along with studies on the terminal

| | Diameter ($\mu$m) | Electric field (V/mm) | Span 80 concentr. (wt%) | $\zeta$ (-) |
|---|---|---|---|---|
| Simulations | $(500, 700, 900)$ | $(300, 500, 700, 900)$ | $(0.0, 0.001, 0.005, 0.016)$ | $[0.15 - 0.80]$ |
| Experiments | $[578 - 902]$ | $[207 - 747]$ | $(0.001, 0.005, 0.016)$ | $[0.15 - 0.65]$ |

Table II: Ranges for parameters used.

velocity. Detailed comparisons between the experimental results and the simulation results are given there. In light of those results, an experimental and a computational parameter study were set up to better understand the effects of surfactants. The aim of the present study is to give a sufficient coverage of the parameter space, and to leverage the combination of simulations and experiments to give a deeper insight into the observed phenomena. In particular we study the initial drop oscillations when the pulse is applied, and the effect the Marangoni force has on the damping of these oscillations. To the best of our knowledge this has not been studied before. The present study was constructed such that we take advantage of the benefits that the computational studies have to offer, namely full control over all parameters and a detailed view of the flow field and surfactant distribution, while at the same time keeping the results directly comparable with the experiments. The data generated by this study are available in the supplementary information.

For the simulations, a combination of three drop radii, four electric field strengths and four surfactant concentrations was chosen, representative of the parameter ranges used in the experiments. The dimensionless field strength $\zeta$ was then computed for each combination, and additional combinations were added to ensure a good coverage of the $\zeta$ values. The values are summarised in table II.

From the experimental point of view, the surfactant concentration is also well-defined at three values (the clean system cannot be reached). The drop radius, on the other hand, is a quantity most difficult to control from one drop to the next, so there is no systematic variation in it. Finally the applied electric field strength is defined from a base value and several fractions of this value, e.g. $(2/6, 3/6, 4/6, 5/6, 6/6)$. This is thus more controlled than the radii. But the base value was varied for different interfacial-tension values, due to a desire to avoid stretching drops beyond their stability limit, as drop destruction necessitates stopping the experiment and cleaning the test cell. The values used in experiments are also summarised in table II.

The results of the simulation-based parameter study are presented in the next section, and the results of the experiment-based parameter study are presented in the subsequent section. Note that the simulations are all done independently, while the experiments are done with several applications of fields of different strength on the same drop. Thus the simulations neglect any hysteresis effects that arise from the hydrodynamics, e.g. if the flow caused by the previous deformation is still significant when the next one commences. Also, since the scaling analysis of the capillary numbers presented in section II indicates that gravity (i.e. the external flow due to falling) is unimportant for the deformations, the simulations are performed with zero gravity.

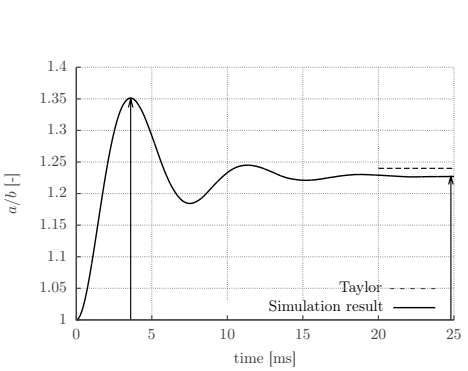
To confirm that these assumptions do not affect the result, at the end of the next section we present results from a simulation with gravity and with an applied increasing voltage pulse as used in the experiments, showing that neither the simplification of zero-gravity nor that of independent deformations has a significant effect on the simulation results.

In the subsequent sections we show plots of the deformation $a/b$, as discussed in section II. During the deformation of a drop, $a/b$ starts as 1, then increases to a peak value, and finally settles at some static value after some oscillations. In figure 4 we plot the time evolution of $a/b$ for one of the cases considered here, as an example. We also show a snapshot of the pressure, velocity and electric fields after 2 ms, which is halfway to the peak deformation. Note that the maximum time in this line plot is the same as the duration of a pulse, so this plot indicates the relaxation towards the new equilibrium of a stretched drop. See also the accompanying movie 1, where a plot like figure 4a from one experiment is shown side-by-side with the high-speed footage of a drop. In our previous work [12] we show detailed comparisons of such plots for experimental and simulation results.

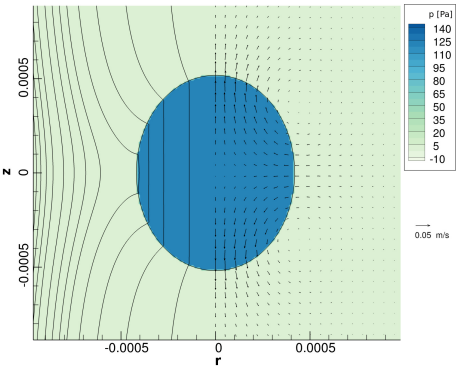
Care has been taken to ensure an exact correspondence of shapes and sizes between all plots in this paper, unless otherwise explicitly stated. Also, the range on the abscissa of the critical electric field $\zeta$ is the same in all plots, except for figure 10 which shows comparison between the simulation and experimental results.

### B. Computational parameter study

In this section we report the results of parameter studies of the deformation as a function of the dimensionless electric field strength. In total 44 cases were studied. The simulations




(a) The deformation, $a/b$, plotted as a function of time. The peak and static values are pointed out by arrows, and the result by Taylor for the static deformation is shown as a dotted line on the right.




(b) Snapshot of the deforming drop after 2 ms, halfway to the peak deformation. On the right side, velocity vectors are shown for every fifth grid point. On the left side, electric field lines are shown, quadratically spaced due to axisymmetry. The colour indicates pressure.

Figure 4: Example case for $\zeta = 0.52$, a 0.9 mm diameter drop is being deformed by a 700 V/mm field without any surfactants present. The field lines

were performed in axisymmetry, using a 241×482 grid covering a $3D \times 6D$ domain. This is about six times larger in each direction than what is shown in figure 4b. As discussed previously we perform the simulations with zero gravity. The initial condition is then a circular drop at rest, with an initial surfactant concentration given by the bulk concentration according to equation (6). The electric field is switched on at $t = 0$; the time it takes in experiments for the voltage to reach its constant value is much smaller than the $\sim 0.5$ ms it takes for the drop to start deforming. See also the supplemental material in movie 2, where we show an animated 3D rendering produced from one of these simulations, namely of the 900 $\mu$m diameter drop in 0.005 wt% Span 80 subjected to a 700 V/mm electric field, corresponding to $\zeta = 0.58$.

The results for the static deformation obtained in the simulations are shown in figure 5. Here the static deformation for each case is shown as a point and compared with the line

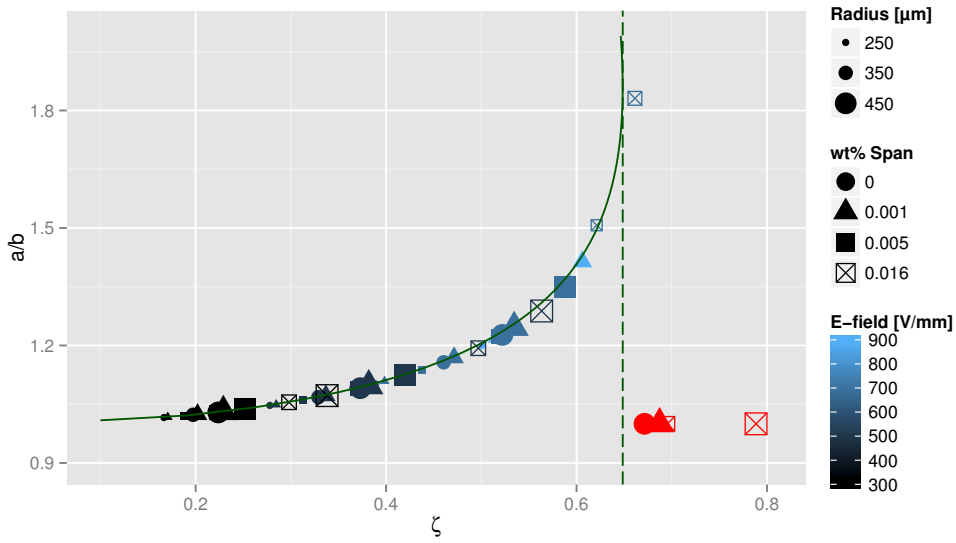



Figure 5: The deformation $a/b$ found in simulations as a function of the dimensionless electric field strength, $\zeta$. Solid line: Taylor theory. Dashed line: static deformation limit. Red points: drop breakup.

which is the Taylor result. The points are colour coded by the electric field. The shape of the points indicates the surfactant concentration, and the size of the points indicates the drop radius. The points shown in red were unstable, i.e. the drop stretched until breakup.

As is seen from figure 5, there are deviations from the Taylor theory, occurring mostly for large values of the dimensionless field strength. It is also seen that the dimensionless parameter $\zeta$ is still a good variable for describing the system in the presence of surfactants. We note that previous simulations by Brazier-Smith [14] also find some slight disagreement with the results by Taylor. This is discussed further in section V.

To further study the effects of increasing surfactant concentration on the oscillation of drops, one may consider the analogy to a damped mass-spring system. In that case, it is more convenient to work with $a-b$ rather than $a/b$. This is because the former is directly related to the magnitude $c_2$ of the fundamental mode of the oscillation, given by the coefficient of the second spherical harmonic, *viz.* $a-b = \sqrt{45/16\pi} \times c_2$. Here we assume that essentially only the second spherical harmonic contributes to the oscillation. Note that under the typical

assumptions in analytical work on drop oscillations, as used e.g. by Lamb [28, pp. 473-475], the temporal evolution of $c_2$ corresponds exactly to the evolution of a damped harmonic oscillator.

Working then with $a - b$ we define the *overshoot* $\Omega$ of an oscillation as

$$\Omega = \frac{(a-b)_{\text{peak}} - (a-b)_{\text{static}}}{(a-b)_{\text{static}}}. \tag{19}$$

This is motivated again by analogy with the damped mass-spring system, where the overshoot of the response to a step forcing has a one-to-one correspondence with the damping ratio [71, p. 172]. This measure of the damping is more accurate here than the standard method of fitting an exponential, since the observed oscillations have few discernible peaks. Using the overshoot is less sensitive to uncertainties and can be used even when just one or two peaks are discernible. Using the formula given in [71] we compute the damping ratio $\lambda$ as

$$\lambda = \sqrt{\frac{\ln(\Omega)^2}{\pi^2 + \ln(\Omega)^2}}. \tag{20}$$

Note that it is more traditional in the context of drop oscillations to work with the damping coefficient $b = \lambda\omega_0$, where $\omega_0$ is the natural frequency, as is done by Lamb [28, p. 474] and by others. This is a bit curious, since their results predict a damping ratio which is directly proportional to the Ohnesorge number, with a proportionality constant depending on the number of the oscillation mode in question; taking e.g. the result by Lamb for the fundamental oscillation mode of a free droplet we obtain $\lambda = 10/\sqrt{3} \times Oh$. For the damping coefficient $b$ the relationship with $Oh$ also includes $\omega_0$.

Even though it is known that the oscillations of a viscous drop immersed in a viscous fluid cannot be described by a simple harmonic oscillator [31], we posit here that $\lambda$ still gives a good measure of how damped the drop oscillations are. Note that it follows from the definition that $\lambda < 1$ corresponds to underdamped oscillations, and that lower values indicate less damped oscillations. See figure 13 in Appendix A where we plot the oscillations for two cases with the same value of $\zeta$ (and thus the same final deformation), but different values of the damping ratio, together with the step responses of harmonic oscillators with the same damping ratios. This plot indicates that $\lambda$ is a useful measure of the damping.

Having defined this damping ratio $\lambda$, we show in figure 6 a plot of $\lambda$ versus $\zeta$ where we connect points with identical drop size and electric field strength. From this plot we may surmise that adding small amounts of surfactant increases the damping significantly, but has

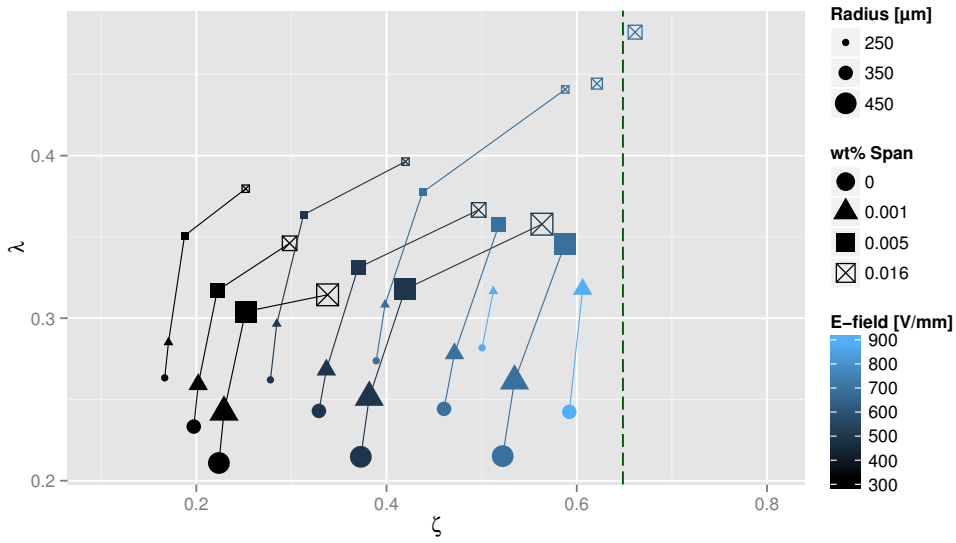



Figure 6: The damping ratio of oscillations, $\lambda$, versus $\zeta$. Dashed line: static deformation limit.

a negligible effect on the static deformation (represented here by $\zeta$), so the lines connecting the 0 wt% and the 0.001 wt% results have steep slopes. On the other hand, adding larger amounts of surfactant significantly increases the $\zeta$ by reducing $\gamma$, so the slopes are flatter. This can be understood when considering that the surfactants play a dual role in the system: adding Marangoni stresses and reducing interfacial tension. From these results we see that when small amounts of surfactant are added, the increase in damping from Marangoni forces is much more significant than the reduction in interfacial tension. When more surfactant is added, the effect of reduced interfacial tension becomes pronounced. We discuss this in more detail in section V, and illustrate the point with detailed plots from the numerical simulations.

Finally, we report the results from a simulation where the drop was falling at terminal velocity and subjected to a rising voltage pulse ( ) matching that used in the experiments. A moving grid procedure was used to keep the falling drop in the centre of the computational domain. The base value of the applied field was 500 kV/m, and the fractions 2/7 to 7/7 of this base value were used.

The surfactant concentration was 0.016 wt% and the drop diameter was 900 $\mu$m. Based on the terminal velocity $u_T$, the capillary number is $Ca_T = \eta_2 u_T/\gamma = 0.004$ and the surface Péclet number is $Pe_T = Du_T/D_\xi = 8.5$. (Recall that $D$ is the drop diameter while $D_\xi$ is the surface diffusion coefficient.) For comparison, the electric capillary number is $Ca_E \in [0.03, 0.3]$ for the increasing field strengths. We may define an electric surface Péclet number, by analogy with the electric capillary number, as $Pe_E = D^2\bar{E}^2\epsilon\epsilon_0/(\eta_2 D_\xi)$, which gives in this case $Pe_E \in [24, 290]$. All in all, these numbers suggest that the external flow is unimportant, which is also what the simulation results indicate.

The simulation results are shown in figure 7. They are very similar to those seen in figure 5, which confirms that neither the external flow nor the previous deformations have a significant influence on the static deformation in the simulations. If the hysteresis effect observed in the experiments described in the next section were caused by hydrodynamic effects or by the surfactant transport, e.g. if the flow field was still influenced by the previous deformation at the start of the next deformation, the hysteresis would also be observed in these simulations. Since it is not, these can be ruled out as likely explanations of the hysteresis.

### C. Experimental parameter study

Several experiments were performed with different drop diameters, field strengths and surfactant concentrations, as described in section IV A. In total, 295 drop deformations were observed, with 8 to 12 observations of each drop and 4 to 6 different voltages applied with both polarities.

The results for the static deformation obtained in the experiments are shown in figure 8 for rising voltage pulse trains ( ) and figure 9 for falling voltage pulse trains ( ). The use of point shapes and sizes match those used in figure 5 for the simulations. In these two plots, the shaded region around the Taylor result indicates the magnitude of the uncertainty in the optical observations, as described in section III A.

It is seen in these two figures that the experiments are also well-described by the parameter $\zeta$, and that the results lie fairly close to the Taylor theory, especially for low deformations. Below $\zeta \sim 0.4$, the deviations are of the same magnitude as the uncertainty in the optical measurements, while above this, they are significant. See also figures 14 and 15 in Appendix

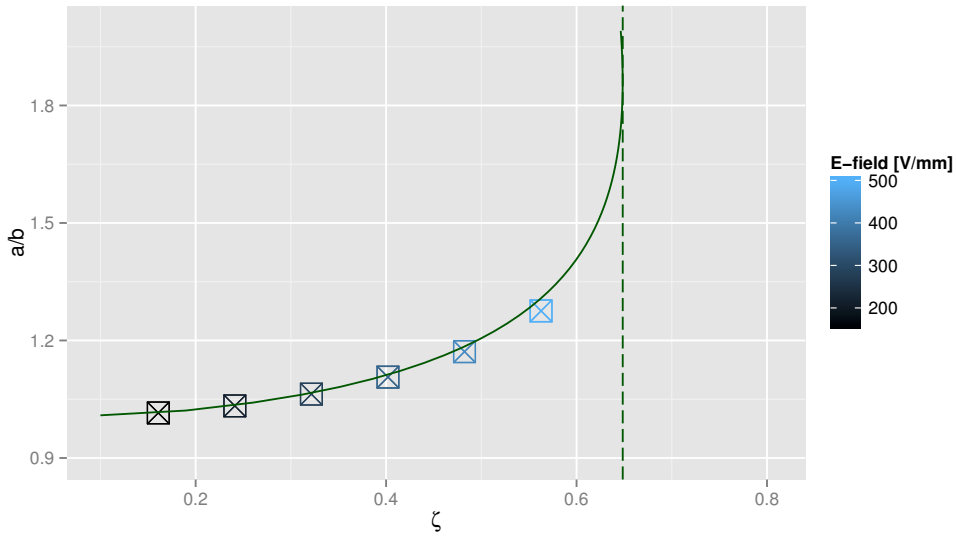



Figure 7: The static deformation $a/b$ versus $\zeta$ for a 900 $\mu$m diameter drop at 0.016 wt% surfactant concentration. This simulation is with a drop falling under gravity and subjected to repeated deformations by a rising voltage pulse.

A where the relative deviation $\Delta$ from the Taylor theory is plotted.

There is a profound difference seen between rising and falling pulse trains in these plots, in that the deviation is positive for the latter, but both positive and negative for the former. (Again, this is also clearly apparent in figures 14 and 15 in Appendix A.) Furthermore it is seen that this difference occurs only for the highest concentration of surfactants.

Peltonen and Yliruusi [40] have reported similar hysteresis effects when repeatedly stretching and compressing an interface between water and hexane with added Span 80, using a Langmuir-Blodgett apparatus. When comparing figure 8 and figure 9 given here, it is noted that for rising voltage pulses, the previous stretchings at small and intermediate field strengths significantly affect the subsequent stretchings, giving a deviation from the Taylor theory that has the opposite sign of that seen in all other cases. Note also that the simulation results (figure 5) show positive deviations for all but one point.

The hypothesis in [40] is that earlier compressions disperse surfactants into the water phase, but it is not readily apparent that this is the case here; if this were so, the large

expansions and compressions of the interface that occur at the beginning of a falling voltage pulse train should significantly affect the subsequent medium and small expansions, but this is not observed.

For the sake of clarity, we remark that the hysteresis seen here is an entirely different phenomenon from the hysteresis studied e.g. by Sherwood [72]. Sherwood considers the hysteresis in the deformation $a/b$ which arises from a finite permittivity ratio, but as is apparent from his figure 2, the phenomenon he discusses requires deformations $a/b \gg 10$, while our deformations are all $a/b < 2$. Thus the hysteresis phenomenon observed herein cannot be attributed to a finite permittivity ratio.

Another interesting phenomenon seen in the numerical results is the effect of surfactant concentration on the damping of oscillations, cf. figure 6. Naturally, it is interesting to see if the experiments show a similar trend. Since the experimental results do not admit the same direct comparison by holding two parameters identical while varying a third, the electric field was binned into 5 intervals with limits at $(200, 400, 500, 600, 700, 800)$ V/mm. No experiments were done with electric fields below 200 or above 800 V/mm. We then considered all results within such a bin and with a given surfactant concentration and joined these into groups, amounting to averaging over the radius. This gave $5 \times 3 = 15$ groups. We omit deformations where the difference between the maximum and the static deformation was smaller than the optical resolution, i.e. we only consider observably underdamped oscillations.

To create a plot like figure 6, the centre-of-mass of each group was computed, connecting centre-of-mass points that represent groups with the same range of electric fields. This corresponds to averaging over the drop radius. (See also figures 16 and 17 in the Appendix for illustrations of these groups and their centres-of-mass.) Plotting this together with the simulation results in figure 10, it is seen that a similar trend is found, in particular for the slopes between the 0.001 and 0.005 wt% results. However, the absolute value of the damping is lower

In this comparison, note that the circular points corresponding to zero surfactant can only be shown for the simulation results, since the system is known to be contaminated even when no Span 80 is added. Note also that in the plot of the simulation results, only those drops with $D \geq 600\mu$m are shown in order to reduce clutter.

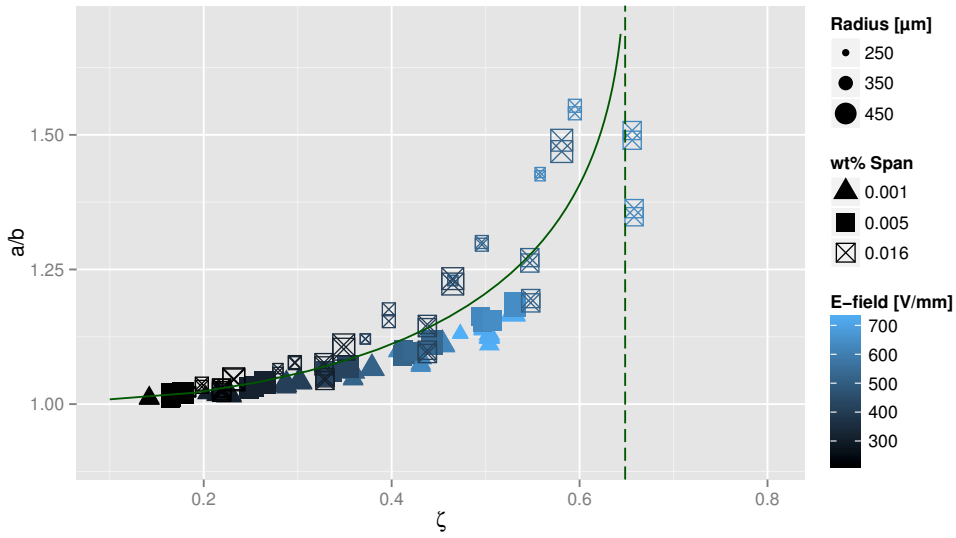



Figure 8: The deformation $a/b$ found in experiments as a function of $\zeta$ for **rising** voltage pulse trains. Shaded region: optical measurement uncertainty.

## V. DISCUSSION

Several features of the results of these studies warrant further comment. First of all, we observe stable solutions slightly beyond the limit predicted by Taylor. One could imagine that this is caused by the effective increase in surface tension as the drop stretches and the surfactant is spread out over a larger area. At the highest surfactant concentration and the largest drop deformations this results in a decrease of about 2% in $\zeta$, which is sufficient to explain the deviation. However, for the lower surfactant concentrations the effect is too small to account for the observed deviation. It is likely that this discrepancy is caused by some of the approximations used by Taylor. Other authors have also pointed out minor disagreements between their results and the Taylor theory. Brazier-Smith [14], using numerical iterations to obtain pressure balance along the entire interface, found a difference which is very similar to that found here.

As for the general agreement with the Taylor theory, it is apparent from the numerical results that the main effect of surfactants on the static deformation is through the reduction

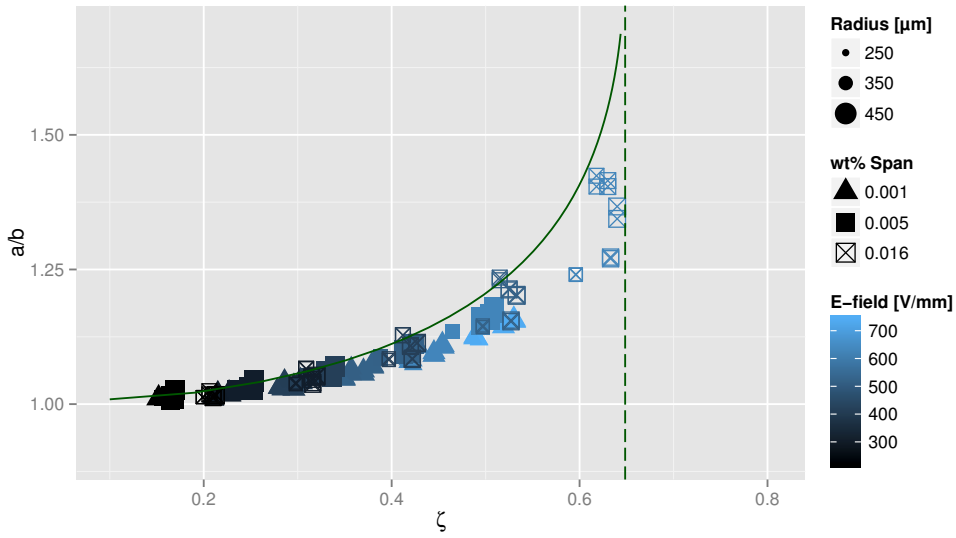



Figure 9: The deformation $a/b$ found in experiments as a function of $\zeta$ for **falling** voltage pulse trains. Shaded region: optical measurement uncertainty.

in equilibrium surface tension. When this is taken into account in the calculation of $\zeta$, the results agree nicely with Taylor's prediction, as shown in figure 5.

For the experimental results, however, there is a clear tendency for smaller deformations than those predicted by the Taylor theory. One possibility is that this discrepancy is caused by a form of interfacial elasticity which is not accounted for here, as discussed earlier.

As illustrated in figure 10, the effect of adding surfactant is two-fold. Small additions increase the damping significantly, while the reduction in interfacial tension is small, so the static deformation is not much affected. Conversely, adding larger amounts of surfactants give a significant decrease in interfacial tension causing a larger deformation, while the increase in damping is less significant. In order to explain this effect, we show in figure 11a plots of the pressure field as well as the vector quantity $\gamma\kappa\boldsymbol{n} - 100 \times \boldsymbol{\nabla}_\iota\gamma$ at the interface for the four different surfactant concentrations considered here. This is for the 0.5 mm diameter drop subjected to a 700 V/mm electric field. We scale the Marangoni force by 100 to accommodate the visualisation, since the curvature $\kappa$ is very large for these small drops. The important thing here is not the absolute value of this vector, but rather the comparison

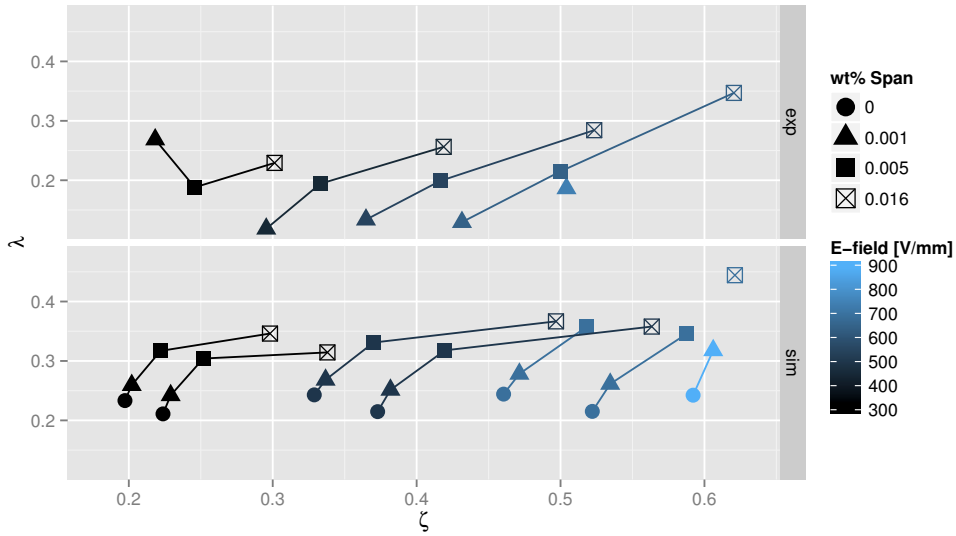



Figure 10: The damping ratio $\lambda$ versus $\zeta$, for the experimental values (top) and the simulation results shown in figure 6 (bottom). The simulation results are only shown for drops with $D \geq 600\mu$m to avoid clutter.

between the four cases in the tangential and normal components. All plots are shown at the same time, $t = 1$ms, which is a little less than halfway to the peak deformation. Around this time, the Marangoni forces are at their largest, since these forces counteract the surfactant maldistribution driving the concentration profile to be uniform at equilibrium. This time is also convenient since the deformation of the drops is very similar at this point in time, while they differ more at later times. Also shown in this plot is the flow field, and the surfactant distribution along the interface in red colour going from the lowest (darkest) to the highest (brightest) concentration along the interface in each case. The interfacial positions are very similar at this early time, which in turn means that both the electric fields and the curvature profiles are also very similar. However, note that the flow is stronger for the drop with the highest surfactant concentration, consistent with the fact that this will be more deformed than the other drops.

In all the three cases where surfactants are present, it is seen that the initial deformation gives an increased surfactant concentration near the equator and a reduced concentration

near the poles. The lowest and highest concentrations $\Gamma_{\min}$ and $\Gamma_{\max}$ for each case with surfactant present are plotted in figure 12 as functions of time. For comparison, the maximum possible interfacial concentration given by fitting the Langmuir EoS equation (7) to the experimental data is 1.31 $\times 10^{-4}$ mol/m$^2$, and all values are well below this.

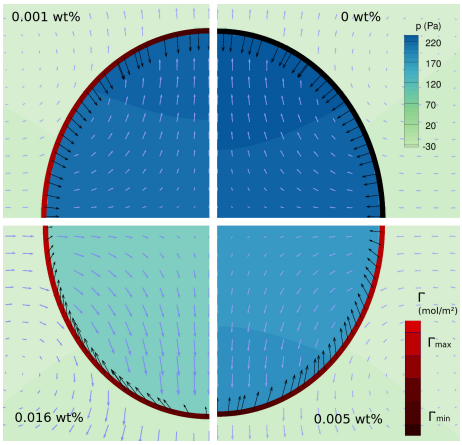
In figure 11b we show the two components of the vector $\gamma\kappa\boldsymbol{n} - 100 \cdot \boldsymbol{\nabla}_\iota\gamma$ plotted as a function of the vertical coordinate $y$, covering here one quadrant of the drop. Inspecting the plot, it is seen that the Marangoni forces increase significantly at the lower surfactant concentrations, while the decrease in interfacial tension is significant mainly for the highest surfactant concentration. These plots confirm the hypothesis put forward to explain the influence of surfactants on the damping which is seen in figure 6.

When it comes to the hysteresis, we have no clear explanation of the observed phenomena. It is evident from the results shown in figure 7 that the simulations cannot explain the hysteresis, even when taking into account the external flow, the history of previous deformations and the Marangoni effect. There is a possibility that incorporating the elastic effects caused by the surfactant could help explain the hysteresis, but this does not seem to be likely *a priori*, since the forces from the elasticity are expected to decrease the deformation, not increase it. Our results do not appear to support the hypothesis in [40], where it is proposed that the hysteresis can be explained by the previous deformations causing surfactants to detach into the water phase. If this were the case, we should observe the hysteresis also for falling voltage pulses, and we do not. If one may speculate, it could be that the small deformations caused by the initial part of a rising pulse train can cause a phase transition in the monolayer of surfactants at the interface. Such phenomena have been reported in the literature [25].

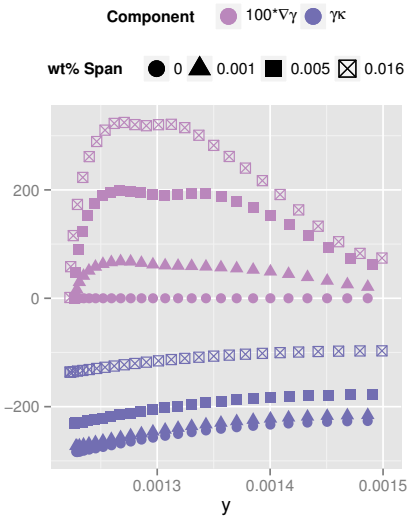
## VI. CONCLUDING REMARKS

We have performed detailed studies of the effect of surfactants on the electrohydrodynamic stretching of water drops in oil at various drop sizes and electric field strengths, covering the full range of dimensionless electric field $\zeta$ from zero to drop breakup. We have compared our results to the classic result by Taylor, which assumes no surfactants present at the drop interface, and predicts the deformation as a function of $\zeta$.

We find that when the equilibrium interfacial tension caused by the surfactant is used in




(a) Comparison of the pressure field (blue/green), the vector $\gamma\kappa\boldsymbol{n} - 100\boldsymbol{\nabla}_\iota\gamma$ (black vectors), the flow field (sky-blue vectors) and the surfactant concentration at the interface (red to black corresponding to the variation shown in figure 12, black for 0 wt% corresponding to $\Gamma = 0$). The quadrants show the four different bulk concentrations considered in this paper.




(b) The quantities $\gamma\kappa$ and $100|\boldsymbol{\nabla}_\iota\gamma|$ along the interface. The abscissa here is the vertical coordinate in (a), such that the left end of this plot corresponds to the pole and the right end to the equator of the drop.

Figure 11: The effect of surfactant concentration on normal and tangential interfacial stress. The plots are for the 0.5 mm diameter drop subjected to a 700 V/mm electric field, at $t = 1$ ms corresponding to the blue vertical line in figure 12. The values of $\zeta$ are 0.39, 0.40, 0.44 and 0.59 in order of increasing surfactant concentration.

the expression for $\zeta$, the system remains well-described by this dimensionless quantity, as expected from figure 12 which shows that the surfactant maldistribution quickly becomes small. For field strengths below $\zeta \sim 0.4$, i.e. deformations below $a/b \sim 1.12$, we have

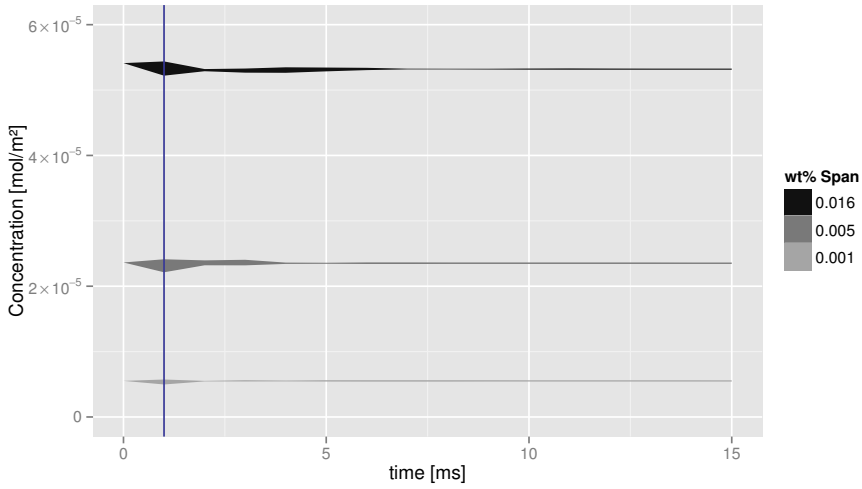



Figure 12: The time evolution of the minimum and maximum interfacial surfactant concentration, shown as the lower and upper edges of shaded bands, for the three cases with surfactants shown in figure 11. The vertical blue line shows the instant at which figure 11 is plotted. Note that all values are well below $\Gamma_\infty = 1.31 \times 10^{-4}$ mol/m$^2$, i.e. within the range of validity of the surfactant EoS.

found only negligible deviations from the Taylor theory when surfactants are added. For field strengths above this we have reported significant deviations of the observed drop deformations, as well as an ability to go slightly beyond the critical stability limit, $\zeta \sim 0.65$, predicted by the Taylor theory without drops breaking up. The deviations from the Taylor theory are larger for experimental than for the simulation results, which could be explained by the interfacial elasticity caused by the surfactant, an effect which is not taken into account in the simulations.

We have shown both by simulations and experiments that the addition of surfactants damps the oscillations induced by a suddenly applied electric field, an effect which may be attributed to the Marangoni effect that arises in the presence of interfacial-tension gradients. We have studied the effects of the surfactant concentration on the damping of oscillations, and found that low concentrations increase the damping significantly while having little effect on the static deformation. On the other hand, the difference between low and high

surfactant concentrations lies mainly in the change of equilibrium surface tension, which affects the static deformation.

Finally we have observed in our experiments a significant hysteresis effect when repeatedly stretching drops at high surfactant concentrations. But these effects are only seen for the case where the applied deformations are first small and then increased, not in the opposite case when the deformations are first large and then decreased. We have investigated whether this effect can be explained just by the hydrodynamics and surfactant transport, and have found this not to be the case. One may speculate that the small initial deformations give the surfactant heads enough room to reorient into an energetically more favorable state, thus making the interface more pliable.

The results presented here add to our understanding of the effects of surfactants on the electrohydrodynamic stretching of conducting drops. But there still remain open avenues of research in this field, in particular on the hysteresis effects. Future work may also consider the interfacial elasticity, to see how this affects the agreement between simulations and experiments, and one may also consider extending the analytical expression by Taylor to include the elasticity.

## ACKNOWLEDGMENTS

We would like to thank Dr. Martin Fossen (SINTEF Petroleum Research) for the measurements of interfacial tension as a function of surfactant concentration, Dr. Velaug Myrseth Oltedal (SINTEF Petroleum Research) for the measurements of bulk viscosity, and Dr. Cédric Lesaint (SINTEF Energy Research) for the measurements of density and for enlightening discussions of the interfacial tension measurements. We would also like to thank Dr. Gunnar Berg (SINTEF Energy Research) as well as Professor Jean-Luc Reboud and Dr. Pierre Atten (G2Elab) and Dr. Erik Bjørklund (Wärtsilä Oil & Gas Systems) for fruitful discussions on the work presented here.

This work was funded by the project *Fundamental understanding of electrocoalescence in heavy crude oils* coordinated by SINTEF Energy Research. The authors acknowledge the financial support from the Petromaks programme of the Research Council of Norway (206976), Petrobras, Statoil and Wärtsilä Oil & Gas Systems.

## APPENDIX: INTERMEDIATE PLOTS

In this appendix we present several plots that further illustrate the steps in the analysis and discussion in the main text. Even more plots are shown in the supplementary information, along with the full data set and the scripts for analysis; see Appendix B for details.

We show one plot in figure 13 which illustrates that the damping ratio as computed from the overshoot is indeed a sensible measure of the damping. Here we show $a - b$ rather than $a/b$, since it is only for $a - b$ that we may compare directly with the damped harmonic oscillator response. In this figure it is seen that the case with a lower damping ratio has more pronounced oscillations. It is also seen that the responses are qualitatively different from the step responses of damped harmonic oscillators. The frequency used for the harmonic oscillator responses are those from the formula given e.g. by Lamb [28, p. 475 eq. 10], illustrating that this expression does not hold here. This is as expected, since that expression is based on several assumptions which are not true in the present case, such as the fluids being weakly viscous. But even if the frequency is altered to make the curves correspond e.g. for the first peak, the curves still do not agree. This is again as expected [31]. Interestingly, however, the time it takes for the oscillations to be damped away is very similar.

Two plots are presented in figures 14 and 15 showing the same results as in figures 8 and 9 save that the relative deviation from the Taylor theory in the static deformation $a/b$, $\Delta = (a/b_{\text{Taylor}} - a/b_{\text{Exp}})/a/b_{\text{Taylor}}$, is plotted. These two figures serve to further illustrate the point that rising pulse trains at the largest surfactant concentration exhibit a significantly different effect than all other observations, namely that the deviation as defined above has the opposite sign.

Two plots are shown in figures 16 and 17 in order to illustrate the analysis that produced the plots in figure 10. First, figure 16 shows the overshoot $\Omega$ plotted versus $\zeta$ for all experimental observations. The points are sorted into 15 groups arising from the combination of the three different surfactant concentrations, and binning the electric field into five groups, with limits at (200,400,500,600,700,800). We average over the radius. The coloured regions in the plot show areas with a 75% probability of finding the points in the different groups. The points in this plot have been shrunk so as to facilitate the inspection. It is seen that

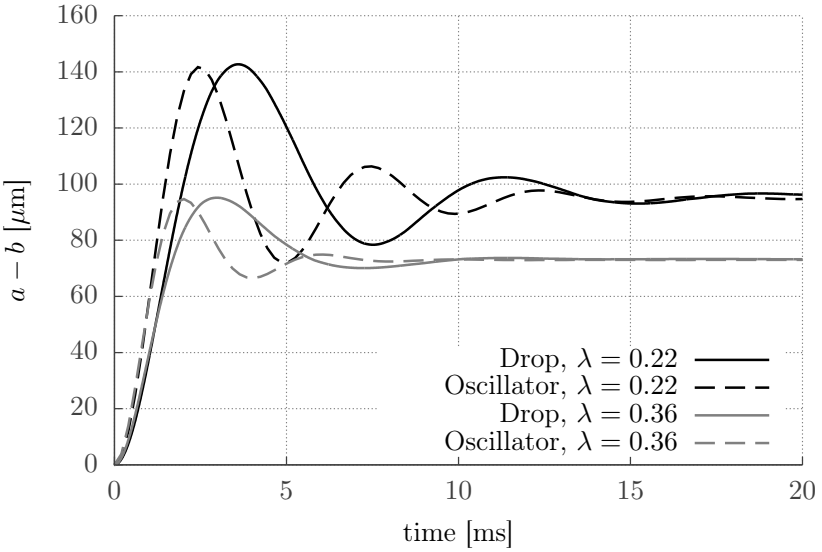



Figure 13: Plots of $a - b$ for two cases with equal values of $\zeta = 0.52$ but different values of the damping ratio. The dashed lines show the step response of a damped harmonic oscillator with the same damping ratio as computed from the overshoot of the simulation results, and with the natural frequency computed from the formula in [28, p. 475].

the binning gives a good representation of the variations in the data.

The second plot, in figure 17, shows the coloured regions in the first plot together with the centre-of-mass of each region, connecting groups with the same radius and electric field strength. From this we can compute the damping for each point and proceed with constructing figure 10.

The data produced by the parameter studies in this paper are permanently stored at Figshare, `http://dx.doi.org/10.6084/m9.figshare.1254343` (note to the reviewers: this link is not yet public, you may instead use the private link `http://figshare.com/s/e27d778e771411e4b1b206ec4b8d1f61`) together with an R Markdown document that uses the `ggplot2` package to construct the majority of the plots in this paper. This means most of the plots in this paper are fully reproducible. The data files are in the CSV format, so they may be opened in any plotting or analysis software. Figures not reproduced by the above document are figures 3, 4, 11a and 13. The line plots in figures 4a and 13 are produced with `gnuplot`. The detailed plots of simulation results in figures 4b and 11a are produced

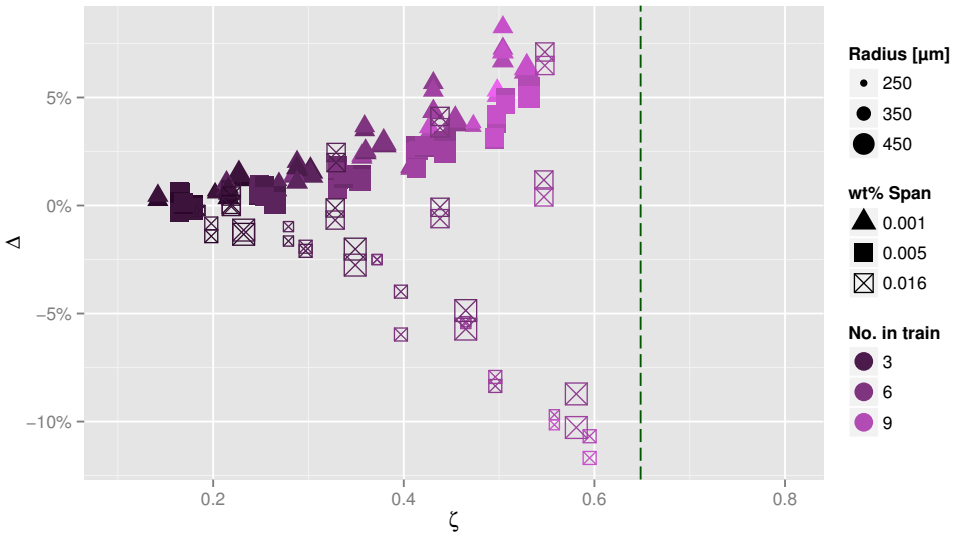



Figure 14: The deviation from the Taylor theory in $a/b$ seen in experiments as a function of the dimensionless electric field strength, $\zeta$, for **rising** voltage pulse trains. Shaded region: optical measurement uncertainty.

with `Tecplot 360 EX`. The 3D renderings in figure 3c and in movie 2 are generated using `Paraview` and `Blender`.

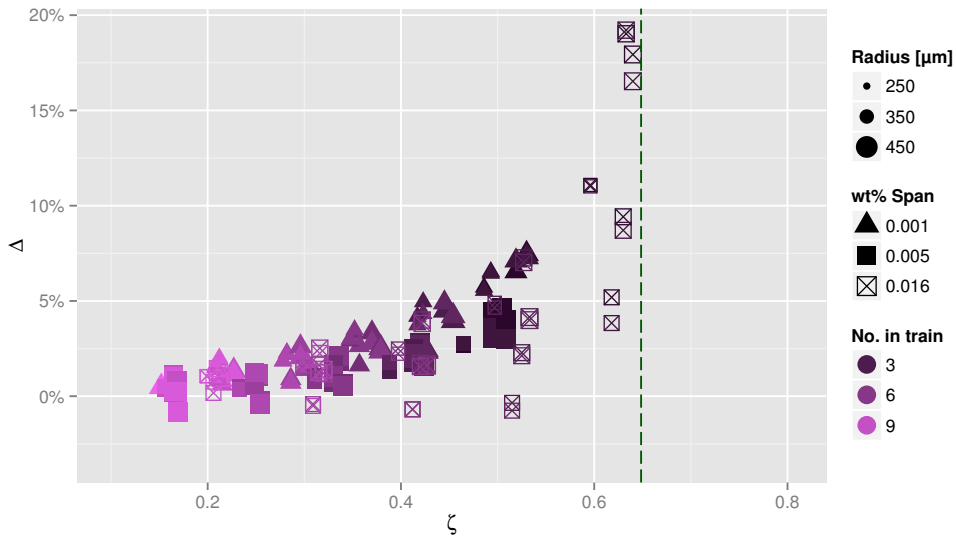



Figure 15: The deviation from the Taylor theory in $a/b$ seen in experiments as a function of the dimensionless electric field strength, $\zeta$, for **falling** voltage pulse trains. Shaded region: optical measurement uncertainty.

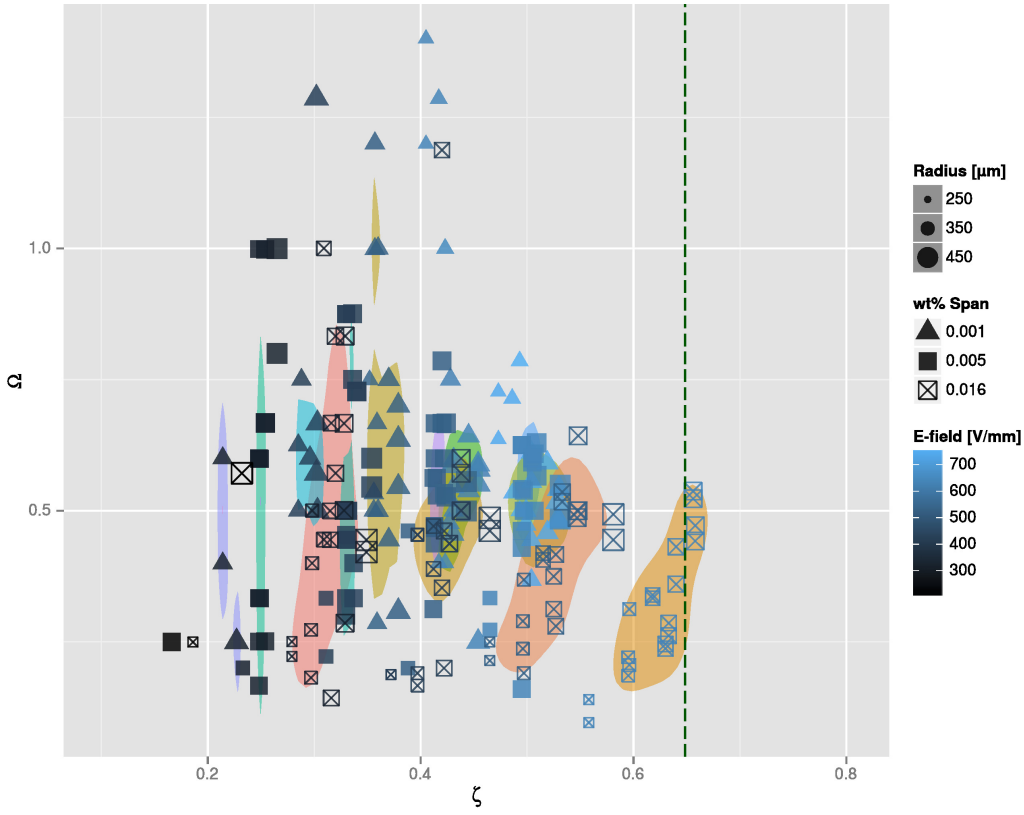



Figure 16: The overshoot $\Omega$ observed in experiments as a function of the dimensionless electric field strength, $\zeta$. The points are binned based on the electric field, and groups with these bins for different surfactant concentrations are computed. The coloured regions indicate the areas where there is a >75% probability of finding points in the various groups. The points in this plot have been shrunk and the sizes cannot be compared with other plots.

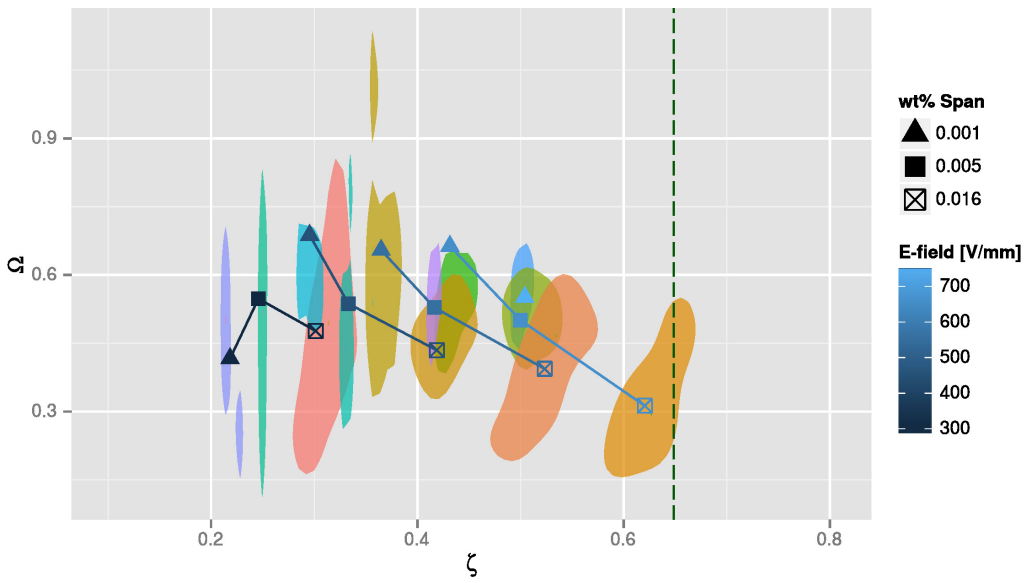



Figure 17: The overshoot $\Omega$ observed in experiments as a function of the dimensionless electric field strength, $\zeta$, for the centre-of-mass of the groups shown in the previous plot. Groups with the same range of electric fields are connected with lines. The colour of the points indicates the mean value of the bins.

[1] J Hadamard, "Mouvement permanent lent d'une sphère liquide et visqueuse dans un liquide visqueux," C. R. Acad. Sci. **152**, 1735–1738 (1911).

[2] W Rybzynski, "Über die fortschreitende Bewegung einer flüssigen Kugel in einem zähen Medium," Bull. Acad. Sci. Cracovie (1911).

[3] Pierre Atten, "Electrocoalescence of water droplets in an insulating liquid," J. Electrostat. **30**, 259–269 (1993).

[4] John S. Eow, Mojtaba Ghadiri, Adel O. Sharif, and Trevor J Williams, "Electrostatic enhancement of coalescence of water droplets in oil: a review of the current understanding," Chem. Eng. J. **84**, 173–192 (2001).

[5] Lars E. Lundgaard, Gunnar Berg, Stian Ingebrigtsen, and Pierre Atten, "Electrocoalescence for oil-water separation: Fundamental aspects," in *Emulsions and Emulsion Stability*, Surfactant Science Series, Vol. 132, edited by Johan Sjöblom (Taylor & Francis, 2006) 2nd ed., Chap. 15, pp. 549–592.

[6] J. R. Melcher and G. I. Taylor, "Electrohydrodynamics: a review of the role of interfacial shear stresses," Ann. Rev. Fluid Mech. **1**, 111–146 (1969).

[7] D. A. Saville, "Electrohydrodynamics: The Taylor-Melcher leaky dielectric model," Ann. Rev. Fluid Mech. **29**, 27–64 (1997).

[8] Jong-Wook Ha and Seung-Man Yang, "Effect of nonionic surfactant on the deformation and breakup of a drop in an electric field," J. Colloid Interf. Sci. **206**, 195 – 204 (1998).

[9] L. Zhang, L. He, M. Ghadiri, and A. Hassanpour, "Effect of surfactants on the deformation and break-up of an aqueous drop in oils under high electric field strengths," J. Petrol. Sci. Eng. **125**, 38 – 47 (2015).

[10] K. E. Teigen, K. Y. Lervåg, and S. T. Munkejord, "Sharp interface simulations of surfactant-covered drops in electric fields," in *Fifth European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2010* (2010).

[11] Herve Nganguia, Y-N Young, Petia M Vlahovska, Jerzy Bławzdziewcz, J Zhang, and H Lin, "Equilibrium electro-deformation of a surfactant-laden viscous drop," Phys. Fluids **25**, 092106 (2013).

[12] Å. Ervik, S. M. Hellesø, S. T. Munkejord, and B. Müller, "Experimental and computational

studies of water drops falling through model oil with surfactant and subjected to an electric field," in *Proceedings of the IEEE 18th International Conference on Dielectric Liquids* (Bled, Slovenia, 2014).

[13] Geoffrey Taylor, "Disintegration of water drops in an electric field," Proc. R. Soc. A **280**, 383–397 (1964).

[14] P. R. Brazier-Smith, "Stability and shape of isolated and pairs of water drops in an electric field," Phys. Fluids **14**, 1–6 (1971).

[15] W. N. Bond and D. A. Newton, "LXXXII. Bubbles and drops and Stokes' law. (Paper 2)," Phil. Mag. **5**, 794—800 (1928).

[16] G. G. Stokes, "On the effect of the internal friction of fluids on the motion of pendulums," Trans. Camb. Phil. Soc. **9**, 8 (1851).

[17] Veniamin G Levich, *Physicochemical hydrodynamics*, 2nd ed. (Prentice-Hall, 1962).

[18] R. Clift, J.R. Grace, and M.E. Weber, *Bubbles, drops and particles* (Academic Press, 1978).

[19] B. Szyszkowski, "Experimentelle Studien über kapillare Eigenschaften der wässerigen Lösungen von Fettsäuren," Z. Phys. Chem. **64**, 385–414 (1908).

[20] Stanislav Samuilovich Dukhin, Günter Kretzschmar, and Reinhard Miller, *Dynamics of adsorption at liquid interfaces: theory, experiment, application*, Vol. 1 (Elsevier, 1995).

[21] Y. Pawar and K. J. Stebe, "Marangoni effects on drop deformation in an extensional flow: The role of surfactant physical chemistry. i. insoluble surfactants," Phys. Fluids **8**, 1738–1751 (1996).

[22] E.H Lucassen-Reynders, A Cagna, and J Lucassen, "Gibbs elasticity, surface dilational modulus and diffusional relaxation in nonionic surfactant monolayers," Colloids Surf. A **186**, 63 – 72 (2001).

[23] Jian-Jun Xu, Zhilin Li, John Lowengrub, and Hongkai Zhao, "A level-set method for interfacial flows with surfactant," J. Comput. Phys. **212**, 590–616 (2006).

[24] EK Sakata and JC Berg, "Surface diffusion in monolayers," Ind. & Eng. Chem. Fundam. **8**, 570–575 (1969).

[25] Francesca Ravera, Michele Ferrari, Eva Santini, and Libero Liggieri, "Influence of surface processes on the dilational visco-elasticity of surfactant solutions," Adv. Colloid Interf. Sci. **117**, 75 – 100 (2005).

[26] George Bidone, "Expériences sur la forme et sur la direction des veines et des courants d'eau

lancés par diverses ouvertures," Memorie della Reale Accademia delle scienze di Torino , 229 (1829).

[27] Lord Rayleigh, "On the capillary phenomena of jets," Proc. R. Soc. **29**, 71–97 (1879).

[28] H. Lamb, *Hydrodynamics*, 6th ed. (Cambridge University Press, Cambridge, 1932) p. 738.

[29] S. Chandrasekhar, "The oscillations of a viscous liquid globe," Proc. London Math. Soc. **s3-9**, 141–149 (1959).

[30] CA Miller and LE Scriven, "The oscillations of a fluid droplet immersed in another fluid," J. Fluid Mech. **32**, 417–435 (1968).

[31] A. Prosperetti, "Free oscillations of drops and bubbles: the initial-value problem," J. Fluid Mech. **100**, 333–347 (1980).

[32] Maria Ronay, "Determination of the dynamic surface tension of liquids from the instability of excited capillary jets and from the oscillation frequency of drops issued from such jets," Proc. R. Soc. A **361**, 181–206 (1978).

[33] Robert E Apfel, Yuren Tian, Joseph Jankovsky, Tao Shi, X Chen, R Glynn Holt, Eugene Trinh, Arvid Croonquist, Kathyrn C Thornton, Albert Sacco Jr, *et al.*, "Free oscillations and surfactant studies of superdeformed drops in microgravity," Phys. Rev. Lett. **78**, 1912–1915 (1997).

[34] Erik Bjørklund, "The level-set method applied to droplet dynamics in the presence of an electric field," Comput. Fluids **28**, 358–269 (2008).

[35] Karl Yngve Lervåg, *Calculation of interface curvatures with the level-set method for two-phase flow simulations and a second-order diffuse-domain method for elliptic problems in complex geometries*, Ph.D. thesis, Norwegian University of Science and Technology, Faculty of Engineering Science and Technology, Department of Energy and Process Engineering (2013).

[36] Knut Erik Teigen and Svend Tollak Munkejord, "Influence of surfactant on drop deformation in an electric field," Phys. Fluids **22** (2010), article 112104.

[37] Shyam Sunder and Gaurav Tomar, "Numerical simulations of bubble formation from a submerged orifice and a needle: The effects of an alternating electric field," Eur. J. Mech. B/Fluids **56**, 97–109 (2016).

[38] Frank M. White, *Fluid Mechanics*, 6th ed. (McGraw-Hill, 2009).

[39] K. Tsujii, *Surface Activity: Principles, Phenomena, and Applications* (Academic Press, 1998).

[40] L. J. Peltonen and J. Yliruusi, "Surface pressure, hysteresis, interfacial tension, and CMC of

four sorbitan monoesters at water–air, water–hexane, and hexane–air interfaces," J. Colloid Interf. Sci. **227**, 1 – 6 (2000).

[41] He Zhao, *An Experimental Investigation of Liquid Droplets Impinging Vertically on a Deep Liquid Pool*, Ph.D. thesis, NTNU (2009).

[42] Robert J. Moffat, "Describing the uncertainties in experimental results," Exp. Therm. Fluid Sci. **1**, 3–17 (1988).

[43] Shi-Yow Lin, Kevin McKeigue, and Charles Maldarelli, "Diffusion-controlled surfactant adsorption studied by pendant drop digitization," AIChE J. **36**, 1785–1795 (1990).

[44] Makio Iwahashi, Atsushi Umehara, Kenichiro Wakisaka, Yasutoshi Kasahara, Hideyuki Minami, Hideyo Matsuzawa, Hideyuki Shinzawa, Yukihiro Ozaki, , and Masao Suzuki, "Effect of cholesterol and other additives on viscosity, self-diffusion coefficient, and intramolecular movements of oleic acid," J. Phys. Chem. B **111**, 740–747 (2007).

[45] Shi-Yow Lin, Ya-Chi Lee, Ming-Wei Yang, , and Hwai-Shen Liu, "Surface equation of state of nonionic $c_m e_n$ surfactants," Langmuir **19**, 3164–3171 (2003).

[46] Knut Erik Teigen and Svend Tollak Munkejord, "Sharp-interface simulations of drop deformation in electric fields," IEEE T. Dielect. El. In. **16**, 475–482 (2009).

[47] Guang-Shan Jiang and Chi-Wang Shu, "Efficient implementation of weighted ENO schemes," J. Comput. Phys. **126**, 202 – 228 (1996).

[48] Guang-Shan Jiang and Danping Peng, "Weighted ENO schemes for Hamilton-Jacobi equations," SIAM J. Sci. Comput. **21**, 2126–2143 (2000).

[49] Alexandre Joel Chorin, "Numerical solution of the Navier-Stokes equations," Math. Comput. **22**, 745–762 (1968).

[50] Van Emden Henson and Ulrike Meier Yang, "BoomerAMG: a parallel algebraic multigrid solver and preconditioner," Appl. Numer. Math. **41**, 155–177 (2000).

[51] H. van der Vorst, "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems," SIAM J. Sci. Comput. **13**, 631–644 (1992).

[52] R.D. Falgout, J.E. Jones, and U.M. Yang, "The design and implementation of hypre, a library of parallel high performance preconditioners," in *Numerical Solution of Partial Differential Equations on Parallel Computers* (Springer-Verlag, 2006) pp. 267–294.

[53] R.D. Falgout, J.E. Jones, and U.M. Yang, *hypre: High Performance Preconditioners*, Lawrence Livermore National Laboratory (2014), `http://www.llnl.gov/CASC/hypre/`.

[54] Satish Balay, William D Gropp, Lois Curfman McInnes, and Barry F Smith, "Efficient management of parallelism in object-oriented numerical software libraries," in *Modern Software Tools for Scientific Computing*, edited by A. Bruaset, E. Arge, and H.P. Langtangen (Springer, 1997) pp. 163–202.

[55] Sigal Gottlieb, David I Ketcheson, and Chi-Wang Shu, "High order strong stability preserving time discretizations," J. Sci. Comput. **38**, 251–289 (2009).

[56] J.L. Guermond, P. Minev, and Jie Shen, "An overview of projection methods for incompressible flows," Comput. Meth. Appl. Mech. Eng. **195**, 6011 – 6045 (2006).

[57] Benjamin Lalanne, Lucia Rueda Villegas, Sébastien Tanguy, and Frédéric Risso, "On the computation of viscous terms for incompressible two-phase flows with level set/ghost fluid method," J. Comput. Phys. **301**, 289 – 307 (2015).

[58] Stanley Osher and James A Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," J. Comput. Phys. **79**, 12 – 49 (1988).

[59] Stanley Osher and Ronald P. Fedkiw, "Level set methods: An overview and some recent results," J. Comput. Phys. **169**, 463 – 502 (2001).

[60] Daniel Hartmann, Matthias Meinke, and Wolfgang Schröder, "The constrained reinitialization equation for level set methods," J. Comput. Phys. **229**, 1514 – 1535 (2010).

[61] D. Adalsteinsson and J. A. Sethian, "The fast construction of extension velocities in level set methods," J. Comput. Phys. **148**, 2 – 22 (1999).

[62] Ronald P. Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher, "A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method)," J. Comput. Phys. **152**, 457–492 (1999).

[63] Myungjoo Kang, Ronald P. Fedkiw, and Xu-Dong Liu, "A boundary condition capturing method for multiphase incompressible flow," J. Sci. Comput. **15**, 323–360 (2000).

[64] Salih Ozen Unverdi and Grétar Tryggvason, "A front-tracking method for viscous, incompressible, multi-fluid flows," J. Comput. Phys. **100**, 25–37 (1992).

[65] Cyril W Hirt and Billy D Nichols, "Volume of fluid (VOF) method for the dynamics of free boundaries," J. Comput. Phys. **39**, 201–225 (1981).

[66] Alain Karma and Wouter-Jan Rappel, "Phase-field method for computationally efficient modeling of solidification with arbitrary interface kinetics," Phys. Rev. E **53**, R3017 (1996).

[67] N. Dubash and A.J. Mestel, "Breakup behavior of a conducting drop suspended in a viscous

fluid subject to an electric field," Phys. Fluids **19** (2007).

[68] Y. Lin, P. Skjetne, and A. Carlson, "A phase field model for multiphase electro-hydrodynamic flow," Int. J. Multiphase Flow **45**, 1–11 (2012).

[69] L. Bai, L. Ni, C. Guo, and Q. Wei, "Numerical analysis of deformation and break-up of aqueous drop of water-in-oil emulsion in high voltage D. C. fields," Chin. J. Appl. Mech. **30**, 76–79 (2013).

[70] J. Zhang, H. He, and G. Huang, "Simulation of droplet deformation in uniform electric field with dissipative particle dynamics approach," Chin. J. Chem. Eng. **65**, 3872–3877 (2014).

[71] K. Ogata, *Modern Control Engineering*, 5th ed. (Prentice Hall, New Jersey, 2009) p. 912.

[72] J. D. Sherwood, "Breakup of fluid droplets in electric and magnetic fields," J. Fluid Mech. **188**, 133–146 (1988).

# Paper 5

# The admissible surfactant distributions and velocities for small falling drops

**Åsmund Ervik**[1]†, **Erik Bjørklund**[2]

[1]Department of Energy and Process Engineering
Norwegian University of Science and Technology (NTNU)
NO-7491 Trondheim, Norway

[2] Wärtsilä Oil & Gas Systems
Solbråveien 10, NO-1383 Asker, Norway

(Received ?; revised?; accepted?. - To be entered by editorial office)

The Hadamard-Rybczynski solution for the flow inside and around a small spherical falling drop is well known. But experimental results for small spherical drops show that these fall slower than predicted by the theory, and instead according to Stokes' result for a falling hard sphere. Increasing the drop size, a transition between these two extremes is found. This phenomenon is due to surfactants present in the system, and study of this has led to the stagnant-cap model. In the present work, we present an alternative model, called the continuous-interface model. In contrast to previous studies using the stagnant-cap model, we do not make use of a surfactant advection-diffusion equation at the interface. Rather, we demonstrate that employing such an equation leads to inconsistent results. Accounting instead for the Marangoni effect through the tangential boundary condition at the interface, and considering also the normal interfacial stress, we are able to solve the Stokes equation analytically for the falling drop with a varying interfacial tension. Some of the solutions thus obtained, e.g. the hovering drop, violate conservation of energy unless energy is provided directly to the interface, for instance by a thermal gradient. In the absence of such an energy input, considering the energy budget of the drop, we show that the terminal velocity is bounded by the Stokes and the Hadamard-Rybczynski results, respectively. To proceed from this result, the continuous-interface model is obtained by considering the balance of forces acting on surfactants at the interface. The resulting expressions predict the functional form of the transition between the hard-sphere and clean-drop results. The model also predicts that the critical radius, below which drops fall like hard spheres, is directly proportional to the *interfacial* surfactant concentration, and thus related to the *bulk* surfactant concentration through a Langmuir isotherm. By analysing experimental results from the literature, we confirm this prediction, thus providing strong arguments for the validity of the proposed model.

**Key words:** Intentionally blank

---

## Nomenclature

| | | |
|---|---|---|
| $C_n^{-1/2}$ | $n$'th Gegenbauer polynomial. | 1 |
| $D$ | Diameter of a drop. | m |
| $E$ | Energy consumption. | W |
| $E_{\mathrm{HS}}$ | Energy consumption of a hard sphere. | W |

† Email address for correspondence: asmunder@pvv.org

| | | |
|---|---|---|
| $E_{\mathrm{I}}$ | Energy consumption in the interface. | W |
| $R$ | Radius of a drop. | m |
| $U$ | Free stream velocity. | m/s |
| $U_{\mathrm{HR}}$ | Terminal velocity of a clean drop. | m/s |
| $U_{\mathrm{HS}}$ | Terminal velocity of a hard sphere. | m/s |
| $\beta$ | Relative viscosity($\mu_1/\mu_2$). | 1 |
| $\delta(\mathbf{x})$ | Dirac delta-function. | 1/m |
| $\kappa$ | Interfacial curvature. | 1/m |
| $\mu$ | Dynamic viscosity of a fluid. | Pa·s |
| $\rho$ | Density of a fluid. | kg/m$^3$ |
| $\sigma$ | Interfacial tension. | N/m |
| $\mathrm{P}_n$ | $n$'th Legendre polynomial. | 1 |
| $\mathrm{e_K}$ | The specific kinetic energy. | J/m$^3$ |
| $\theta$ | Polar angle, measured from the z-axis. | rad |
| $\mathbf{T}$ | Cauchy stress tensor. | N/m$^2$ |
| $\mathbf{f}_{\mathrm{b}}$ | External acceleration. | m/s$^2$ |
| $\mathbf{f}_{\mathrm{I}}$ | Interfacial force. | N |
| $\mathbf{g}$ | Gravitational acceleration. | m/s$^2$ |
| $\mathbf{x}_{\mathrm{I}}$ | Material points on the interface. | m |
| $p_{0i}$ | Reference pressure in the $i$'th fluid phase. | Pa |
| $r$ | Radial distance from droplet centre. | m |

## 1. Introduction

A single falling (or rising) drop is one of the simplest two-phase flow configurations, and has been under scrutiny since the dawn of fluid mechanics research. Many of the early studies were focused on drops impacting a pool of water, such as the works by Worthington (1876) and by Reynolds (1875). Stokes (1851) was the first to give an analytical solution for the flow at low Reynolds number ($Re$) around a solid sphere falling at terminal velocity. Then Hadamard (1911) and Rybzynski (1911) independently published the analytical solution for the flow inside and around a clean spherical drop falling at low $Re$. This has later been extended by various authors to account for the presence of surfactants, under various assumptions, as will be discussed in the following.

The case of liquids with surfactants may seem to be of lesser interest than the case of clean fluids. But experimentally observed terminal velocities of small drops do not match the Hadamard-Rybczynski result, but rather the Stokes result for the vast majority of combinations of "clean" fluids, see e.g. the work by Nordlund (1913); Lebedev (1916); Silvey (1916); Bond (1927); Bond & Newton (1928). In the latter work, a distinguished jump was found in the terminal velocity, going from the Stokes result to the Hadamard-Rybczynski result as the drop radius was increased. This has been confirmed in later experiments, e.g. by Griffith (1962).

It is noteworthy that even Hadamard acknowledges the fact that his expression does not agree with experimental results, in the closing words of his 1911 paper, where he refers to disagreement between the expression and some (at that time) unpublished experimental results:

> *La formule (III) présente, avec les résultats expérimentaux obtenus quant à présent (et encore inédits), de notables divergences. Il semble donc, jusqu'á nouvel ordre, que, dans les cas étudiés, les hypothèses classiques dont nous sommes parti doivent être modifiées.*

In fact there are extremely few published works that are able to obtain terminal

velocities for very small drops matching the Hadamard-Rybczynski result, and then only for quite singular fluid combinations. Examples include molten lead drops in liquid beryllium trioxide (Volarovich & Leont'yeva 1939), or liquid mercury drops in highly purified glycerine (Frumkin & Bagotskaya 1947). There are a few studies where the authors have gone to great pains to purify more ordinary fluid systems, but these have been limited to $Re > 10$, see e.g. Thorsen *et al.* (1968); Edge & Grant (1972). That one is able to obtain agreement with the Hadamard-Rybczynski result for small drops only when at least one of the fluids in question are chemically quite different from ordinary liquids, supports the hypothesis that amphiphilic surfactants which occur naturally in even highly purified organic liquids cause the aberrant behaviour typically observed.

In later years, attention towards surfactants and their role in systems both man-made (e.g. in various foods) and natural (e.g. in our lungs) has increased considerably. As an example, it is recognised that surfactants play a dominant role in the stability of emulsions (Lucassen-Reynders 1996), whether this is stability is desired (as in mayonnaise) or not (as in a water-crude oil emulsion). Surfactants act both to slow down the sedimentation of drops, and to prevent the coalescence of drops in a separation process.

Several authors have considered the extension of the Hadamard-Rybczynski analytical result to account for the presence of surfactants. Prominent examples include the work by Frumkin & Levich (1947), Savic (1953)†, Davis & Acrivos (1966), Griffith (1962) and Sadhal & Johnson (1983). This body of work incorporates both experimental data, in the form of correlations, and exact results under various assumptions; see Clift *et al.* (1978, Chapter II.D) for a review.

A prominent feature in these works is the assumption of a stagnant cap, i.e. that the surfactant is accumulated at the top of a falling drop, such that the interface is immobile in this region and free to move on the rest of the drop. This assumption is based on photographic evidence gathered for larger drops. An example is the photograph in the paper by Savic (1953), reproduced here in Figure 1, which is often taken as *prima facie* evidence for the stagnant cap model.

In this paper, we will extend the derivation by Chang & Berg (1985) of the analytical solution for the terminal velocity of a low $Re$ circular drop with an arbitrary surfactant (hence interfacial tension) distribution, taking here also the normal interfacial stresses into account. We show that this leads to a plethora of solutions, some of which are clearly unphysical (in the absence of an external energy input), such as a hovering drop. By appealing to the conservation of energy, we show that the physically admissible terminal velocities are bounded from below by the Stokes result for hard spheres, and from above by the Hadamard-Rybczynski result.

We proceed to supplement this with a simple model for the forces acting on surfactant molecules at the interface, giving an expression for the transition in terminal velocity between the two extremal values. This expression depends on the properties of the surfactant in question. From the theory we predict that for a given surfactant, the critical radius $R_c$ below which drops fall like hard spheres, should be proportional to the *interfacial* surfactant concentration. To confirm this prediction we determine the critical radii for the different *bulk* surfactant concentrations considered in the experiments performed by Griffith (1962). We demonstrate that these critical radii, when plotted against the *bulk* surfactant concentration, all collapse on a single Langmuir isotherm. Since the Langmuir isotherm relates the interfacial and the bulk surfactant concentration,

† The report by Savic has not been available electronically in the past; however we were informed by the National Research Council of Canada that the copyright on it has expired, and have thus made a scanned copy available at `http://archive.org/download/mt-22/savic.pdf`

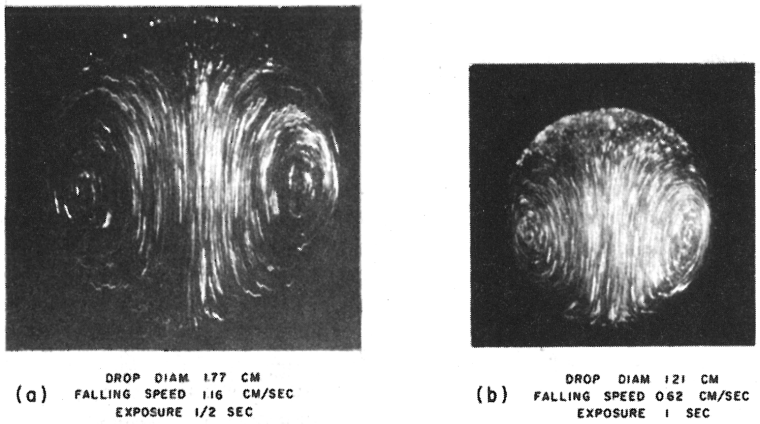



FIGURE 1. From the flow visualisation studies by Savic. Image showing a falling water drop in castor oil, with complete internal circulation, (a), and a stagnant cap, i.e. a downwards shift of the internal flow pattern, (b). Reproduced from Savic (1953) with permission from the National Research Council of Canada.

this confirms the prediction by the present model, which we call the continuous-interface model.

We then proceed to discuss the existing versions of the stagnant-cap model and to compare these with the model presented in this work. We demonstrate in detail that the stagnant-cap model leads to paradoxical results. As an example, consider Equation 7-270 on page 497 in the book by Leal (2007), which reads

$$\nabla_s \cdot (\mathbf{u}_s \Gamma) \approx 0 \tag{1.1}$$

where $\Gamma$ is the surfactant concentration. This equation follows from the assumption that the surfactant is insoluble, and that the interfacial Péclet number is $Pe_s = 2R|\mathbf{u}_s|/D_s \gg 1$ where $\mathbf{u}_s$ is the velocity at the interface and $D_s$ is the interfacial diffusivity of the surfactant. From this, the classic stagnant cap result is obtained, namely that the part of the interface where the surfactant is found has $\mathbf{u}_s = 0$ (this is the stagnant cap), and the rest of the interface has $\Gamma = 0$. But if $\mathbf{u}_s = 0$ in the stagnant cap region where the surfactant is located, then $Pe_s = 0 \not\gg 1$. Thus the result violates the initial assumption, and so it is not self-consistent.

It is notable that the result obtained in this work for the interfacial tension distribution along the drop interface has the same functional form as the result obtained in the classic analysis e.g. by Levich (1962). But unlike Levich we do not assume the variation in surfactant concentration to be small, and since the present work avoids the use of a surfactant advection-diffusion equation on the interface, in contrast to previous approaches, we are able to obtain simultaneous analytical solutions to the flow and the interfacial tension distribution. This has been a major obstacle in previous work, as noted e.g. by Leal (2007):

> *It is not generally possible to obtain analytic solutions of the resulting problem because of the complexity of the surfactant transport phenomenon and the coupling between surfactant transport and fluid motion.*

In closing, we argue that the present model is more appropriate for interfacially active agents which are amphiphilic molecules, while the stagnant-cap model may indeed remain appropriate for dispersed particulates which adsorb at the interface and thus modify the boundary conditions of the problem.

## 2. Theoretical results

### 2.1. *Governing equations*

The flow field $\mathbf{u}$ of an incompressible viscous Newtonian fluid is governed by the Navier-Stokes equations on the form

$$\nabla \cdot \mathbf{u} = 0, \tag{2.1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{\nabla p}{\rho} + \frac{\mu}{\rho}\nabla^2 \mathbf{u} + \mathbf{f}_\mathrm{b}. \tag{2.2}$$

Here $p(\mathbf{x})$ is the pressure field and $\mathbf{f}_\mathrm{b}$ is some external acceleration, such as gravity. As it stands, this system of equations is closed when fluid properties and initial and boundary conditions are given.

The system can be extended to two fluids by specifying an interface that separates fluid $_1$ with properties $\rho_1, \mu_1$ from fluid $_2$ with properties $\rho_2, \mu_2$, as well as two dynamic interfacial relations related to the interfacial tension $\sigma$. For the case of a drop or bubble we will mark the internal properties with $_1$ and the external properties with $_2$.

In order to have closure of Equation (2.2) with this extension, one also needs the following interfacial relations for two-phase flow:

$$[\![\mathbf{u}]\!] = \mathbf{0}, \tag{2.3}$$

$$\mathbf{t} \cdot [\![\mathbf{T}]\!] \cdot \mathbf{n} = -\mathbf{t} \cdot \nabla\sigma, \tag{2.4}$$

$$\mathbf{n} \cdot [\![\mathbf{T}]\!] \cdot \mathbf{n} = \kappa\sigma. \tag{2.5}$$

Here the jump in a quantity across the interface is denoted by $[\![\text{-}]\!]$, $\mathbf{n}$ is the normal and $\mathbf{t}$ is the tangent vector, and $\mathbf{T}$ is the stress tensor. We choose the normal vector to point out from a drop, and the jump is then given by e.g. $[\![\mu]\!] = \mu_2 - \mu_1$, i.e. the difference between the bulk and the drop value. In the case of a spherical droplet with zero velocity field, Equation (2.5) reduces to the Young-Laplace relation for the pressure difference across an interface ($\Delta p = 2\sigma/R$). The Marangoni force comes in through Equation (2.4), and the functional form of the coefficient of interfacial tension along the drop interface, $\sigma$, is also needed for closure.

In this paper, the case of one spherical droplet falling in an unbounded domain will be considered. In this case, it is natural to introduce the following characteristic properties:

$$x^* = R, \quad u^* = U, \quad t^* = R/U, \quad p^* = \sigma/R, \tag{2.6}$$

giving the following non-dimensional Navier-Stokes equations:

$$\nabla \cdot \mathbf{u} = 0, \tag{2.7}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \frac{1}{Re}\left[\nabla^2 \mathbf{u} + \frac{Re}{We}\left(Eo\,\mathbf{f} - \nabla p\right)\right]. \tag{2.8}$$

Here, $Re$ denotes the Reynolds number, $We$ is the Weber number and $Eo$ represents the Eötvös number. The Reynolds number is given by $Re = \rho_2 U R/\mu_2$ as it is customary to use the continuous fluid properties in the dimensionless groups. This dimensionless number gives the ratio of inertial forces to viscous forces. The Weber number is given by $We = \rho_2 U^2 R/\sigma$ and gives the ratio between inertial forces and interfacial tension forces. Lastly, the Eötvös number, $Eo = \rho_2 g R^2/\sigma$, gives the ratio between the body forces and the capillary forces.

In what follows, $Re$, $We$ and $Eo$ are all assumed to be small. The assumption of $Re \ll 1$ and of steady state flow simplifies the Navier-Stokes equation to the steady Stokes

equation:

$$\nabla^2 \mathbf{u} + \frac{Re}{We}\left(Eo\,\mathbf{f} - \nabla p\right) = \mathbf{0}. \tag{2.9}$$

When $We$ is small, $We \ll 1$, the forces due to interfacial tension determine the shape of the interface through minimising the interfacial energy, which results in a spherical drop. As demonstrated by the Hadamard-Rybczynski result, the spherical falling drop is an exact solution to the Stokes equation. Furthermore, as demonstrated by Kojima *et al.* (1984) and in subsequent work (see Stone (1994, Chapter 6) for a review), perturbations away from the spherical shape for a drop falling at low $Re$ will either relax back toward the spherical shape, or form instabilities as elongated tails.

Lastly, the assumption of a small Eötvös number means that the body forces are small compared to the capillary forces, and hence will not alter the spherical shape of the droplet, but rather induce an acceleration on the droplet as a rigid body. Taylor & Acrivos (1964) considered the deviation from a spherical shape and found that this is $O(Re^2)$, i.e. very small when $Re$ is small. It follows that the assumption of a spherical drop is not a significant restriction, and so this assumption is ubiquitous in the literature on the falling drop at low $Re$ with surfactants (Savic 1953; Levich 1962; Griffith 1962; Davis & Acrivos 1966; Sadhal & Johnson 1983; Leal 2007).

### 2.2. *Spherical droplet in a quiescent liquid*

We will now consider a spherical droplet in a gravitational field surrounded by a quiescent liquid with which the droplet is immiscible. For a perfectly clean interface, the stationary solution is given by the Hadamard-Rybczynski solution. We proceed to let the interfacial tension vary along the interface and investigate the solutions obtained when accounting for the Marangoni force. We will follow in the steps of the analysis of Chang & Berg (1985), but we will also include the interfacial conditions for normal stresses. The appropriate boundary conditions are then given by Equations (2.3)–(2.5).

We employ a spherical coordinate system $(r, \theta, \phi)$ fixed at the centre-of-mass of the droplet, with polar angle $\theta$ measured from the positive z-axis. For convenience we will at times refer to the axes of the Cartesian coordinate system, which has positive x-axis corresponding to $\theta = \pi/2, \phi = 0$. The situation is cylindrically symmetric, so the azimuthal angle $\phi$ is redundant. The normal vectors in the $r$ and $\theta$ directions are denoted by $\mathbf{e}_r$ and $\mathbf{e}_\theta$, respectively. For a spherical drop, the normal (i.e. radial) velocity is zero at the interface, and the velocity far away from the droplet is given by

$$\mathbf{u} = U\cos(\theta)\,\mathbf{e}_r - U\sin(\theta)\,\mathbf{e}_\theta, \tag{2.10}$$

where $U$ is the uniform velocity at infinity. The general solution for the stream functions outside and inside the droplet are found to be

$$\Psi_2 = \sum_{n=2}^{\infty} \left(A_n r^n + B_n r^{1-n} + D_n r^{2+n} + E_n r^{3-n}\right) C_n^{-1/2}, \tag{2.11}$$

$$\Psi_1 = \sum_{n=2}^{\infty} \left(F_n r^n + G_n r^{1-n} + H_n r^{2+n} + I_n r^{3-n}\right) C_n^{-1/2}, \tag{2.12}$$

where the velocity components are related to the stream function by

$$u_r = \frac{1}{r^2 \sin(\theta)} \frac{\partial \Psi}{\partial \theta}, \tag{2.13}$$

$$u_\theta = -\frac{1}{r \sin(\theta)} \frac{\partial \Psi}{\partial r}. \tag{2.14}$$

Requiring a uniform velocity at infinity gives

$$A_2 = -U, \tag{2.15}$$
$$A_n = 0 \quad \forall n \geq 3, \tag{2.16}$$
$$D_n = 0 \quad \forall n \geq 2, \tag{2.17}$$

and the assumption that the velocity is bounded at the origin gives

$$G_n = 0 \quad \forall n \geq 2, \tag{2.18}$$
$$I_n = 0 \quad \forall n \geq 2. \tag{2.19}$$

The vanishing normal velocity at the interface ($r = R$) gives

$$B_2 = R^3 U - R^2 E_2, \tag{2.20}$$
$$B_n = -R^2 E_n \quad \forall n \geq 3, \tag{2.21}$$
$$H_n = -\frac{F_n}{R^2} \quad \forall n \geq 2. \tag{2.22}$$

The final kinematic boundary condition, the continuity of the velocity field across the interface, gives

$$E_2 = \frac{3}{2} RU - RF_2, \tag{2.23}$$
$$E_n = -R^{2n-3} F_n \quad \forall n \geq 3. \tag{2.24}$$

Both stream functions can now be expressed through one common set of coefficients

$$\Psi_2 = \left( -Ur^2 - \left( \frac{R^3}{2} U - R^3 F_2 \right) \frac{1}{r} + \left( \frac{3}{2} RU - RF_2 \right) r \right) C_2^{-1/2} + \sum_{n=3}^{\infty} \left( \frac{R^{2n-1}}{r^{n-1}} - \frac{R^{2n-3}}{r^{n-3}} \right) F_n C_n^{-1/2}, \tag{2.25}$$
$$\Psi_1 = \sum_{n=2}^{\infty} \left( r^n - \frac{r^{n+2}}{R^2} \right) F_n C_n^{-1/2}, \tag{2.26}$$

where $R$ is the radius of the droplet.

The dynamic interfacial conditions will now be used to determine the last coefficient and the interfacial tension as a function of the polar angle. Since the Legendre polynomials form a complete orthonormal basis for any periodic function, we may write

$$\sigma = \sum_{n=0}^{\infty} \sigma_n \mathrm{P}_n(\eta), \tag{2.27}$$

where $\eta = \cos(\theta)$ and $\mathrm{P}_n$ is the $n$'th Legendre polynomial. The normal stress condition (Equation (2.5)) in spherical coordinates can be written as

$$[\![-p]\!] + 2[\![\mu \frac{\partial}{\partial r} \left( \frac{1}{r^2} \frac{\partial \Psi}{\partial \eta} \right)]\!] = \frac{2}{R} \sigma, \tag{2.28}$$

giving the following relations between $F_n$ and $\sigma_n$

$$\sigma_0 = \frac{R}{2}\left(p_{01} - p_{02}\right), \tag{2.29}$$

$$\sigma_1 = \frac{3}{4}\mu_2 U - \frac{1}{2}\left(\rho_1 - \rho_2\right) g R^2 + \left(\frac{3}{2}\mu_2 + 3\mu_1\right) F_2, \tag{2.30}$$

$$\sigma_n = 6R^{n-2}\left(\frac{\mu_i}{2\left(n-1\right)} + \frac{\mu_o}{2n}\right) F_{n+1} \quad \forall n \geq 2. \tag{2.31}$$

Similarly, the shear stress condition (Equation (2.4)), containing the Marangoni force, gives the relations

$$\sigma_1 = -\frac{3}{2}\mu_2 U + 3\left(\mu_2 + \mu_1\right) F_2, \tag{2.32}$$

$$\sigma_n = 2R^{n-2}\frac{2n-1}{n\left(n+1\right)-6}\left(\mu_1 + \mu_2\right) F_{n+1} \quad \forall n \geq 2. \tag{2.33}$$

The simultaneous solution to Equations (2.29)–(2.31) and Equations (2.32)–(2.33) is given by

$$\sigma_0 = \frac{R}{2}\left(p_{01} - p_{02}\right), \tag{2.34}$$

$$\sigma_1 = \frac{9}{2}\mu_1\left(U - U_{\mathrm{HS}}\right) + \frac{1}{2}\mu_2\left(6U - 9U_{\mathrm{HS}}\right), \tag{2.35}$$

$$\sigma_n = 0 \quad \forall n \geq 2, \tag{2.36}$$

$$F_2 = \frac{3}{2}\left(U - U_{\mathrm{HS}}\right), \tag{2.37}$$

$$F_n = 0 \quad \forall n \geq 3, \tag{2.38}$$

where $p_{0i}$ is the reference pressure in the respective phases and

$$U_{\mathrm{HS}} = \frac{2\Delta\rho|\mathbf{g}|R^2}{9\mu_2}, \tag{2.39}$$

where $\Delta\rho = \left(\rho_1 - \rho_2\right)$, is the Stokes result for the terminal velocity of a hard sphere. By insertion, one finds that the expression for $\sigma_1$ is zero if $U$ is replaced by the solution given by Hadamard and Rybczynski, which is consistent with the assumption of two clean fluids.

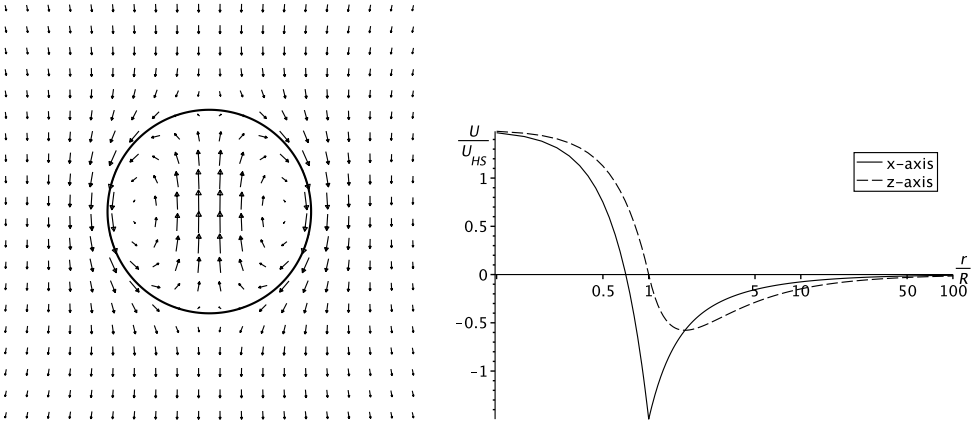
The resulting expressions for the stream functions can now be given by

$$\Psi_1\left(r,\theta\right) = \frac{3}{4}\left(U - U_{\mathrm{HS}}\right)\left(r^2 - \frac{r^4}{R^2}\right)\sin^2\left(\theta\right), \tag{2.40}$$

$$\Psi_2\left(r,\theta\right) = \frac{1}{2}\left(-Ur^2 + \frac{3}{2}U_{\mathrm{HS}}Rr + \left(U - \frac{3}{2}U_{\mathrm{HS}}\right)\frac{R^3}{r}\right)\sin^2\left(\theta\right). \tag{2.41}$$

One may also notice that the internal stream function is identically equal to zero if $U$ is replaced by $U_{\mathrm{HS}}$. This shows that if the droplet is falling with the same velocity as a hard sphere, the Marangoni forces will balance the shear forces from the external fluid, resulting in a uniform velocity inside the droplet equal to the droplet velocity.

The above analysis does not give any restrictions on the velocity of the droplet, other than the demand of keeping the Reynolds number low. An exotic case would be that of a hovering droplet, meaning that the Marangoni forces balance the forces induced by gravity. In the absence of an energy input e.g. from a temperature gradient (Young *et al.* 1959) or from an asymmetric release of surfactants (Masoud & Stone 2014), both of




(a) The vector field $\mathbf{u}$ around the hovering droplet.

(b) Signed magnitude of the velocity along the z- and x-axis, $\theta = 0$ and $\theta = \pi/2$ respectively.

FIGURE 2. The velocity field in the case of a hovering drop (terminal velocity equal to zero). In (a) the vector field is shown close to the drop. In (b) line plots of the velocity along the polar (z-axis) and the azimuthal (x-axis) directions. It is seen that the velocity field decays to zero far away from the drop. Note that the sharp kink in (b) is caused by the drop interface.

which are interesting systems in their own right, a hovering drop will clearly violate the conservation of energy.

The expression for the stream function shows that there is a non-zero velocity field in this case. This is obvious from the fact that in the presence of Marangoni forces, the viscous stress tensor cannot be zero both inside and outside the droplet, and hence there must be gradients in the velocity field. In Figure 2 we plot (a) the vectors of the velocity field around the hovering drop and (b) the decay of the velocity field far away from the drop. This case corresponds of course to $U = 0$, so the coordinate systems of the drop and the laboratory coincide.

To proceed, one may consider the energy balance in this system, in order to pick physically acceptable solutions. The energy equation for creeping flow can be written as:

$$\dot{e}_K = \frac{\partial}{\partial t}\left(\rho\frac{u^2}{2}\right) = \mathbf{u}\cdot(\nabla\cdot\mathbf{T}) + \rho\mathbf{u}\cdot\mathbf{f}_b + \mathbf{u}\cdot\mathbf{f}_I(r,l)\delta\left(\mathbf{x}-\mathbf{x}_I(r,l)\right)\mathrm{d}r\,\mathrm{d}l \tag{2.42}$$

$$= \nabla\cdot(\mathbf{u}\cdot\mathbf{T}) - \mathbf{T}\colon\nabla\mathbf{u} + \rho\mathbf{u}\cdot\mathbf{f}_b + \mathbf{u}\cdot\int_\Gamma \mathbf{f}_I(r,l)\delta\left(\mathbf{x}-\mathbf{x}_I(r,l)\right)\mathrm{d}r\,\mathrm{d}l, \tag{2.43}$$

where $\delta(\mathbf{x}-\mathbf{x}_I)$ is a Dirac delta-function which is singular at the interface and $r$ and $l$ is the parametrisation of the interface. The first term on the right of Equation (2.43) is the energy flux passing through a fluid interface and the second term is the energy dissipation in a fluid element. The third term is the energy provided by the body force term, while the last term is the energy dissipated in the interface due to the action of the surfactants. For the sake of brevity we will refer to this dissipation as "energy consumption" (or "energy production" in the opposite case), even though energy can of course not be consumed or produced.

We will now look at the energy balance of the interface itself. This is achieved by integrating Equation (2.43) over a volume just enclosing the interface (see

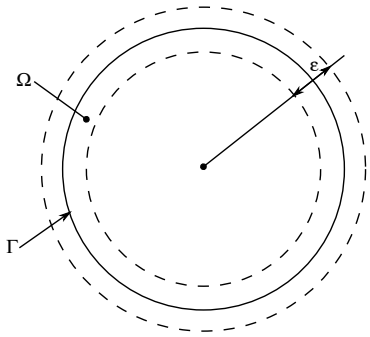



FIGURE 3. The integration domain $\Omega$ of thickness $\epsilon$ around the droplet interface $\Gamma$.

Figure 3):

$$0 = \int_\Omega \nabla\cdot(\mathbf{u}\cdot\mathbf{T})\,\mathrm{d}\mathbf{x} - \int_\Omega \mathbf{T}\colon\nabla\mathbf{u}\,\mathrm{d}\mathbf{x} + \int_\Omega \rho\mathbf{u}\cdot\mathbf{f}_\mathrm{b}\,\mathrm{d}\mathbf{x} + \int_\Omega \mathbf{u}\cdot\int_\Gamma \mathbf{f}_\mathrm{I}\delta(\mathbf{x}-\mathbf{x}_\mathrm{I})\,\mathrm{d}S\,\mathrm{d}\mathbf{x} \quad (2.44)$$

After applying Gauss' theorem and letting $\epsilon$ approach zero (following Hansen (2005)), one obtains

$$0 = \underbrace{-\oint \mathbf{u}_2\cdot\mathbf{T}_2\cdot\mathbf{n}\,\mathrm{d}S}_{E_1} + \underbrace{\oint \mathbf{u}_1\cdot\mathbf{T}_1\cdot\mathbf{n}\,\mathrm{d}S}_{E_2} + \underbrace{\oint (\mathbf{u}_1\cdot\mathbf{t})(\mathbf{t}\cdot\nabla\sigma)\,\mathrm{d}S}_{E_3}, \quad (2.45)$$

where we have labelled the terms for reference in Figure 4, and where

$$\mathbf{u}\cdot\mathbf{T}\cdot\mathbf{n} = -pu_r + \mu\left(2u_r\frac{\partial u_r}{\partial r} + u_\theta\frac{\partial u_\theta}{\partial r} - \frac{u_\theta^2}{r} + \frac{u_\theta}{r}\frac{\partial u_r}{\partial r}\right) \quad (2.46)$$

in spherical coordinates. We have also used an inward pointing normal vector so that energy flux into the interface is positive. Since there is no net movement of the drop, the integration over the droplet of the body force term in Equation (2.43) will be zero. Thus Equation (2.45) shows that the energy consumption in the interface together with the energy dissipation in the droplet at stationary conditions must equal the energy flux into the droplet interface from the surrounding fluid side.

The three terms in Equation (2.45) are plotted individually in Figure 4. The figure shows that there is an interval where the interface is consuming energy to keep the constant velocity. Since we are not providing external energy to the interface, this is the only allowed velocity interval. By setting the third term in Equation (2.45) to zero, a second order equation for the velocity gives:

$$U = U_\mathrm{HS} \quad \vee \quad U = 3\frac{\mu_1+\mu_2}{3\mu_1+2\mu_2}U_\mathrm{HS}, \quad (2.47)$$

as the bounding interval. So, the permissible solutions for a viscous sphere falling at steady state in a gravitational field surrounded by a quiescent liquid under the influence of Marangoni forces are bounded by the Stokes solution for the hard sphere and the Hadamard-Rybczynski solution for clean liquids. Note again the contrast here with the stagnant-cap model, where these two bounds are assumed *a priori*.

To compare this result directly to the stagnant-cap model, one may consider the terminal

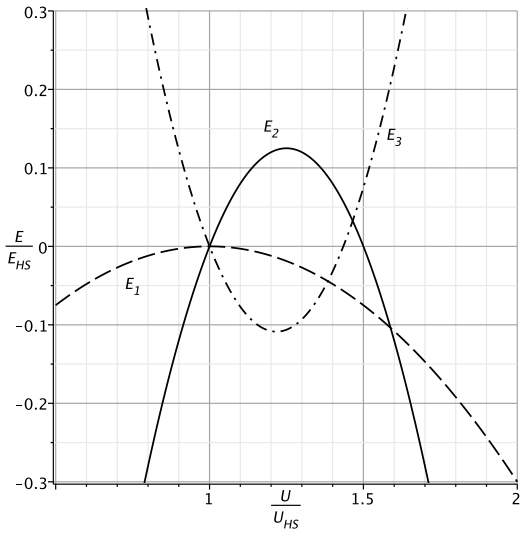



FIGURE 4. Energy consumption for the falling droplet. Negative values indicate energy consumption, while positive values are energy production. The term $E_3$ is the second term on the right-hand side of Equation (2.45), viz. $E_3 = E_1 - E_2$, and we see clearly that the interval in which the interface consumes energy is bounded by the Stokes and the Hadamard-Rybczynski terminal velocities, cf. Equation (2.47). Outside these limits, the interface produces energy, which is unphysical.

velocity of the droplet given as a function of the interfacial tension,

$$U = \frac{3(\mu_1+\mu_2)}{3\mu_1+2\mu_2}U_{\mathrm{HS}} - \frac{2\sigma_1}{9\mu_1+6\mu_2} \tag{2.48}$$

$$= U_{\mathrm{HR}} - \frac{2\sigma_1}{9\mu_1+6\mu_2}, \tag{2.49}$$

where the reader is reminded that the interfacial tension is $\sigma = \sigma_0 - \sigma_1\cos(\theta)$ and that $U_{\mathrm{HR}}$ is the Hadamard-Rybczynski velocity. This gives the following expression for the drag force on the droplet:

$$\mathrm{F_D} = \frac{4\pi\mu_2 UR}{1+\beta}\left[\frac{3}{2}\beta + 1 + \frac{\sigma_1}{3\mu_2 U}\right], \tag{2.50}$$

where $\beta$ is the ratio of the inner and outer fluid viscosity. In comparison, the drag force obtained with the stagnant cap model can be written as

$$\mathrm{F_D} = \frac{4\pi\mu_2 UR}{1+\beta}\left[\frac{3}{2}\beta + 1 + f_1\left(f_2^{-1}\left(\frac{1+\beta}{\mu_2 U}\sigma_\Delta\right)\right)\right], \tag{2.51}$$

where $\sigma_\Delta$ is the difference between the maximum and minimum value of interfacial tension as defined in Sadhal & Johnson (1983), and $f_1$ and $f_2$ are trigonometric functions of the cap angle (Hatanaka *et al.* 1988). Davis & Acrivos (1966) assumed that $\sigma_\Delta$ was limited by a constant value $\Pi^*$, making the argument in $f_2^{-1}$ approach zero when the droplet radius (hence the terminal velocity) increases. One then obtains the desired behaviour with the drag force approaching that for clean droplets.

### 2.3. *The continuous-interface model*

Proceeding from this result, we will derive a mechanical interface model which links the interfacial concentration of surfactants to the coefficient of interfacial tension. To achieve

this we will use arguments from molecular considerations, and link the dynamic equations directly to the Marangoni force.

It is assumed that the surfactant molecules are subjected to a force field, $\mathbf{f}$, and that their action on each other due to thermal fluctuations is governed by a Wiener process, i.e. a force given by $\sigma\mathbf{F}_s$ where $\sigma$ is a scaling parameter for the normalised stochastic Wiener function $\mathbf{F}_s$. It is customary (Giona *et al.* 2004) to model the fluid friction on each molecule by a Stokesian force term, $\mathbf{f} = \psi\mathbf{u}$, where $\psi$ is a friction constant and $\mathbf{u}$ is the velocity of the species surrounding the molecule in question. This leads to Brownian motion, whose stochastic behaviour in the diffusion-controlled regime is governed by the Langevin equation (Giona *et al.* 2004)

$$\frac{d\mathbf{x}}{dt} = -\frac{\mathbf{f}}{\psi} - \frac{\sigma}{\psi}\mathbf{F}_s. \tag{2.52}$$

This equation corresponds to a Fokker-Plank equation which is the macroscopic advection diffusion equation (Castiglione *et al.* 1999),

$$\frac{\partial\Gamma}{\partial t} + \nabla\cdot\mathbf{j} = D_s\nabla^2\Gamma, \tag{2.53}$$

where $\Gamma$ is the interfacial concentration of the species in question, $D_s = 1/2(\sigma/\psi)^2$ is the interfacial diffusion coefficient, and $\mathbf{j} = \mathbf{f}/\psi\Gamma$ is the flux. This advection-diffusion equation is the typical starting point for modelling the transport of surfactants on the droplet interface (Levich 1962; Leal 2007).

Considering this equation, inconsistency again arises. In the situation where the surfactants create a gradient in the conjuration, causing the Marangoni force to retard the droplet to the Stokes terminal velocity, the hydrodynamic equations show that the interfacial velocity of the liquid is zero (using a coordinate system fixed to the droplet centre-of-mass). Inserting a zero velocity into the advection diffusion equation, the stationary solution has to be constant. That is, the concentration of surfactants is uniform over the droplet, giving a uniform coefficient of interfacial tension, and hence contradicting the assumption of a gradient in interfacial tension.

The preceding outline of the derivation of the transport equation identifies the origin of this inconsistency. The prediction of uniform surfactant concentration at zero interfacial velocity by the advection-diffusion equation can be traced back to the assumption of a Stokesian interaction force between the surfactants and the surrounding liquid, since this clearly leads to a vanishing interaction force when the surrounding liquid velocity is reduced to zero. It should be noted here that the assumption of a Stokesian interaction force is necessary in order to obtain an advection-diffusion equation. If the interactions are not described by such a force, one cannot obtain an advection-diffusion equation.

This leads us to conclude that the surfactant transport cannot be modelled via a diffusion-advection equation based on the interfacial velocity. However, since the macroscopic equations demand a continuity in the velocity field across the interface, it is natural to demand that the surfactants should move with the same velocity as the surrounding liquid. It is also required that the forces transferred to the surfactants are governed by the Marangoni force (which is not a given when using an advection-diffusion equation). If one proceeds from these considerations by considering the force balance, one may simultaneously require that both the interfacial force corresponds to the Marangoni force through a body force term, and that the velocity field is equal to the interfacial velocity, *viz.*

$$\Gamma\frac{\partial\mathbf{u}_\mathrm{s}}{\partial t} + \Gamma(\mathbf{u}_\mathrm{s}\cdot\nabla)\mathbf{u}_\mathrm{s} = -\nabla\Pi + \mu_\mathrm{s}\nabla^2\mathbf{u}_\mathrm{s} + \mathbf{f}_\mathrm{M}, \tag{2.54}$$

where $\mathbf{u}_\mathrm{I}$ is the interfacial velocity field, $\mu_\mathrm{s}$ is a viscosity term in the surfactant layer, and $\mathbf{f}_\mathrm{M}$ is the Marangoni force. The pressure term $\nabla\Pi$, where $\Pi$ is the interfacial pressure, accounts for the surfactant-surfactant interaction term. The simplest model of such interactions is the ideal gas law,

$$\Pi = k\Gamma, \tag{2.55}$$

a model which goes back to the classic paper by Langmuir (1917). At stationary conditions, expressing the interfacial velocity field in the spherical coordinate system, Equation (2.54) becomes

$$\Gamma U_\theta^2 \cos(\theta)\sin(\theta) = -\frac{k}{R}\frac{\partial\Gamma}{\partial\theta} + \frac{\mu_s U_\theta}{R^2}\frac{\partial}{\partial\theta}\left(\frac{1}{\sin(\theta)}\frac{\partial}{\partial\theta}\left(\sin^2(\theta)\right)\right) + \frac{\sigma_1}{R}\sin(\theta), \tag{2.56}$$

which has the solution

$$\Gamma(\theta) = e^{\frac{\alpha\cos(2\theta)}{4}}\left(C - \beta\sqrt{\frac{\pi}{2\alpha}}e^{\frac{\alpha}{4}}\operatorname{erf}\left(\sqrt{\frac{\alpha}{2}}\cos(\theta)\right)\right), \tag{2.57}$$

where $\alpha = U^2/k$, $\beta = \sigma/k - 2\mu_s U_\theta/Rk$, and $C$ is a constant of integration. One should notice that

$$\lim_{\alpha\to 0}\Gamma(\theta) = C - \beta\cos(\theta), \tag{2.58}$$

which corresponds to neglecting the convective term. It is then straight-forward to obtain that the coefficient of integration is given by the average interfacial concentration of surfactants, $C = \Gamma_\mathrm{avg}$. In the following it is assumed that the convective term is negligible, and thus Equation (2.58) gives the surfactant concentration at the interface. In addition, it is assumed that the viscous term is small compared to the coefficient of interfacial tension, which corresponds to low interfacial concentration of surfactants. One then obtains

$$\Gamma(\theta) = \Gamma_\mathrm{avg} - \frac{\sigma_1}{k}\cos(\theta). \tag{2.59}$$

This equation links the interfacial concentration of surfactants to the Marangoni force acting on the surfactant molecules by the use of the Gibbs-Marangoni relation in the transport equation for the surfactants. It should be noted that Equation (2.54) does not make any assumptions regarding the functional form of the coefficient of interfacial tension.

Furthermore, it is readily apparent from Equation (2.58) that $\Gamma$ as a function of $\theta$ is symmetric about $\Gamma_\mathrm{avg}$, and that the minimum is given by $\Gamma_\mathrm{avg} - \beta$, which implies that

$$\sigma_1 \leq k\Gamma_\mathrm{avg} + \frac{2\mu_s U_\theta}{R}. \tag{2.60}$$

Positing that there exists a maximum value for the repulsive force between the surfactants, such that if Equation (2.54) would require a higher value, the surfactant would be released from the interface and dissolve into the bulk phase, Equation (2.59) shows that $\Gamma$ must be less than some maximum value $\Gamma_\infty$. $\Gamma_\infty$ is known as the maximum packing concentration in the surfactant literature. Inserting $\Gamma_\infty \geq \Gamma$ into Equation (2.59) one obtains

$$\Gamma_\mathrm{avg} \leq \Gamma_\infty - \sigma_1/k, \tag{2.61}$$

giving

$$\sigma_1 \leq \frac{k\Gamma_\infty}{2}. \tag{2.62}$$

In arriving at these expressions, it was implicitly assumed that there is no limit to the

forces each surfactant molecule can absorb from the surrounding liquids. In reality, the surfactant molecules will bend and twist if they are subjected to large stresses. Taking this into account, i.e. requiring that each molecule can at most absorb a force of magnitude $F_\infty$, one obtains a restriction on the Marangoni shear stress $\tau = \sigma_1 \sin(\theta)/R$, namely $\tau/\Gamma \leq F_\infty$. In addition to this comes the requirement that $\Gamma \leq \Gamma_\infty$ as discussed previously.

Writing it out in full, this expression for the maximum interfacial shear stress is

$$\max_\theta \frac{\tau}{\Gamma(\theta)} = \max_\theta \frac{\frac{1}{R}\sigma_1 \sin(\theta)}{\Gamma_{\text{avg}} + \frac{\sigma_1}{k}\cos(\theta)} \leq F_\infty, \tag{2.63}$$

or equivalently

$$\max_\theta \sigma_1 \left( \frac{\sin(\theta)}{R} - \frac{F_\infty}{k}\cos(\theta) \right) \leq F_\infty \Gamma_{\text{avg}}, \tag{2.64}$$

giving a restriction on the rate of change of the interfacial tension,

$$\sigma_1 \leq \frac{k\Gamma_{\text{avg}}}{\sqrt{1 + \left(\frac{k}{F_\infty R}\right)^2}} \leq k\Gamma_{\text{avg}}, \tag{2.65}$$

which upon insertion of Equation (2.61) yields

$$\sigma_1 \leq \frac{k\Gamma_\infty}{1 + \sqrt{1 + \left(\frac{k}{F_\infty R}\right)^2}} \leq \frac{k\Gamma_\infty}{2}. \tag{2.66}$$

Using Equation (2.48) to eliminate $\sigma_1$ from Equation (2.65), one obtains now an expression for the lowest terminal velocity allowed for a drop of radius $R$. Normalising this by the terminal velocity of a hard sphere, $\chi = U/U_{\text{HS}}$, gives

$$\chi \geq \frac{3(\mu_1 + \mu_2)}{3\mu_1 + 2\mu_2} - \frac{2}{9\mu_1 + 6\mu_2} \frac{k\Gamma_{\text{avg}}}{U_{\text{HS}} \sqrt{1 + \left(\frac{k}{F_\infty R}\right)^2}}. \tag{2.67}$$

At this point it is convenient to introduce the viscosity ratio $\beta = \mu_1/\mu_2$, as well as the quantity $R_c = k/F_\infty$. This critical radius $R_c$ is the largest droplet radius such that the forces from the surfactants on the liquid are large enough to retard the droplet to the Stokes terminal velocity ($\chi = 1$). We denote the drop radius normalised by the critical radius as $x = R/R_c$. Using the fact that this expression should become 1 at $x = 1$, i.e. that drops with the critical radius fall like hard spheres, we obtain $\Gamma_{\text{avg}}/R_c = \sqrt{2}\Delta\rho g/3F_\infty$, and the previous equation simplifies to

$$\chi \geq \frac{3(\beta + 1)}{3\beta + 2} - \frac{\sqrt{2}}{3\beta + 2} \frac{x^{-2}}{\sqrt{1 + x^{-2}}}. \tag{2.68}$$

Notice that in the expression for $\Gamma_{\text{avg}}/R_c$ preceding this equation, the maximum force a surfactant molecule can absorb, $F_\infty$, is a material constant for the surfactant. This means that for a given surfactant, the average interfacial concentration is directly proportional to the critical radius. Thus, in low surfactant-concentration experiments, one may use the critical radius as a measure of equilibrium interfacial concentration of surfactants, $\Gamma_{\text{avg}}$. The reader is reminded that $\Gamma_{\text{avg}}$ can be related to the bulk concentration of surfactants, $C$, by a Langmuir isotherm

$$\Gamma_{\text{avg}} = \Gamma_\infty \frac{aC}{1 + aC} \tag{2.69}$$

where $a$ is a constant.

Recalling the proof in Section 2.2 that a drop cannot fall slower than a hard sphere of equal radius, since this violates the conservation of energy, we obtain the final expression for the relative velocity $\chi$ as

$$\chi(x) \geq \begin{cases} 1 & \text{if } x \leq 1, \\ \frac{3(\beta+1)}{3\beta+2} - \frac{\sqrt{2}}{3\beta+2} \frac{x^{-2}}{\sqrt{1+x^{-2}}} & \text{if } x > 1. \end{cases} \tag{2.70}$$

Notice that this expression is continuous at $x = 1$, but the derivative is discontinuous at this point, cf. Figure 6. Notice also that when $R \gg R_c$, i.e. $x^{-2} \ll 1$, the expression for $x > 1$ approaches the Hadamard-Rybczynski result. Thus the inequality must be replaced by equality in the $x \gg 1$ limit. In the $x \to 1$ limit, equality is also required since it is observed that the drops fall like hard spheres. The simplest expression which is correct in both these limits is obtained by replacing the inequality with equality for the entire expression, *viz.*

$$\chi(x) = \begin{cases} 1 & \text{if } x \leq 1, \\ \frac{3(\beta+1)}{3\beta+2} - \frac{\sqrt{2}}{3\beta+2} \frac{x^{-2}}{\sqrt{1+x^{-2}}} & \text{if } x > 1. \end{cases} \tag{2.71}$$

This is the prediction of the continuous-interface model for the transition in terminal velocity as a function of drop radius.

## 3. Discussion

In 1953, Savic (1953) introduced the stagnant cap model (SCM) as an explanation of the experimental results obtained by Bond (1927); Bond & Newton (1928). The SCM incorporates the effect of surfactants through a rigid cap where a no-slip boundary condition is used. It is obvious that this will lead to a terminal velocity of droplets bounded by the Stokes velocity and the Hadamard-Rybczynski velocity. In later works several iterations of the SCM have been proposed(Griffith 1962; Davis & Acrivos 1966; Harper 1973; Sadhal & Johnson 1983), and presently two different versions exist, namely the model proposed by Griffith (1962) and the model proposed by Davis & Acrivos (1966). The Griffith approach is based on calculating the cap angle from a criterion based on the average interfacial pressure difference, while Davis & Acrivos employ a local criterion based on the capillary tension and the interfacial shear forces.

Hatanaka *et al.* (1988) give a review of the experiments performed by Bond & Newton (1928) and Griffith (1962) and compare the two versions of the SCM with the experimental results. Hatanaka *et al.* show that the model of Davis & Acrivos gives better agreement with the experiments performed by Bond & Newton (1928), while the Griffith model gives better agreement with the experimental results performed by Griffith himself. It appears that the difference between the experimental results by Griffith and those of Bond & Newton is too large to be governed by the same mechanism. Note here that while the experiments due to Bond & Newton (1928) use fluids which are assumed to be pure, in the experiments by Griffith (1962) a surfactant is deliberately added at known bulk concentrations. In general, one considers the experiments performed by Griffith (1962) to be more reliable, since the experimental setup there is better controlled, taking advantage of developments in our understanding of chemistry and fluid mechanics, as well as developments in experimental equipment, not available at the time of Bond & Newton (1928).

Figure 5 shows one set of experiments performed by Griffith (1962), extracted from

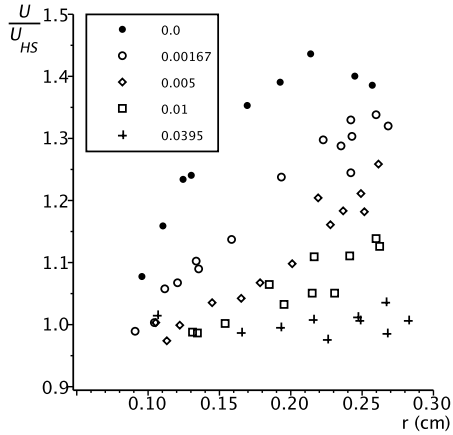



FIGURE 5. Droplet terminal velocities for different drop sizes and at different surfactants concentrations (Griffith (1962, Fig. 9)). The concentration is given in $\mathrm{kg/m^3}$ and the highest concentration corresponds to 35.4 ppm.

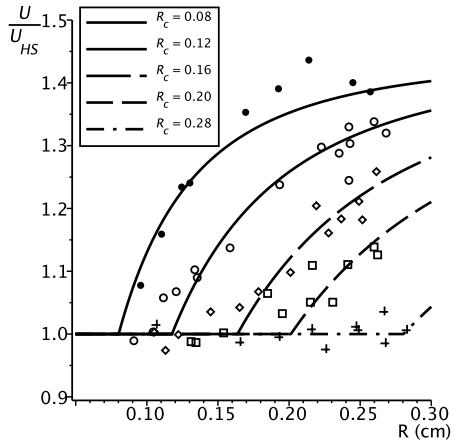



FIGURE 6. Nonlinear curve fit of Equation (2.71) to the raw data by Griffith (1962).

Figure 9 in his work. The figure shows the results from an experiment with drops of ethylene glycol with Aerosol 61 surfactant falling in a reservoir of a mineral oil, using different concentrations of surfactant. The results are plotted as the relative velocity $\chi = U/U_{HS}$ versus the drop radius. Notice that the base fluids employed by Griffith are not free of surface active agents, indicated by the results without any added surfactant showing the same trend of approaching the hard sphere terminal velocity as the radius decreases.

By fitting Equation (2.71) to these data points, we may calculate the critical radius $R_c$ below which $\chi = 1$ in the experiments performed by Griffith. This is shown in Figure 6, where the obtained values of $R_c$ for each of the five concentrations is shown in the legend. Note that in theory, if perfectly pure fluids were used, $R_c \to 0$ as $C \to 0$. This is not the case for these experiments.

As outlined in the previous section, the proposed continuous-interface model predicts that the critical radius is directly proportional to the interfacial surfactant concentration, which can again be related to the bulk concentration through the Langmuir isotherm Equation (2.69). This means the critical radius is also related to the bulk concentration

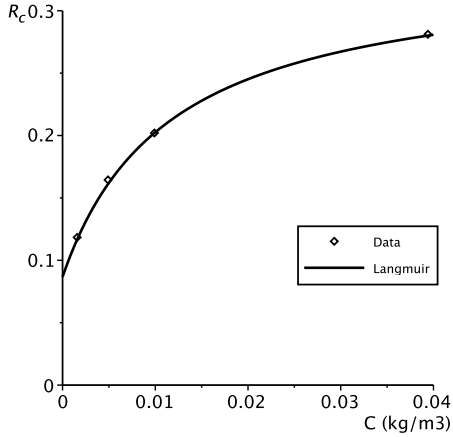



FIGURE 7. $R_c$ obtained from the fitting of Equation (2.71) to the experimental data, plotted against bulk surfactant concentration, together with a Langmuir isotherm fitted to these points.

via a Langmuir isotherm, i.e. we can write $R_c = R_c(C)$. Notice, however, that the isotherm must be modified to account for the fact that surfactants are still present in the system at $C = 0$, i.e. $R_c(C = 0) \neq 0$. By replacing the concentration in Equation (2.69) with $C' = C + C_{\text{base}}$, we obtain an isotherm with two unknown parameters, $a$ and $C_{\text{base}}$. Fitting this to the experimental data, as shown in Figure 7, it is seen that the critical radii all collapse to the obtained Langmuir isotherm. This confirms the prediction made by the proposed continuous-interface model.

To complete this discussion, we wish to point out that the stagnant cap model is likely a good model when the surface-active agents interact like hard particles. To see this, consider the explanation of the stagnant cap model in terms of the Marangoni force, as attempted by Sadhal & Johnson (1983). If the Marangoni stress balance is valid, then arguable the normal stress balance should also be satisfied. In the SCM, none of the above stress balances are used as boundary conditions. It is then only possible (given the uniqueness of solutions to the Stokes equation) to satisfy both stress balances if one of them can be written in terms of the boundary conditions used. This is obviously not possible, and hence the normal stress balance cannot be satisfied.

It is therefore natural to conclude that the SCM is limited to situations where interfacially active components interact like particles and form a solid cap. In the continuous-interface model presented here, both stress balances are used in the boundary conditions and therefore, in contrast to the SCM, it can be used when the interfacially active components are amphiphilic molecules.

If the SCM is reasonable for interfacially active components that act as particles, one might want to re-examine experimental evidence for the SCM that is obtained using particle-based flow visualisation methods. If it is the addition of particles for the flow visualisation that causes the stagnant cap which is observed, these experiments are obviously not representative for the more general situation of molecular contaminants causing small liquid drops to fall like hard spheres. It would perhaps be possible to test this hypothesis by performing NMR-based velocity field measurements (Han *et al.* 2001) of small drops falling both with and without tracer particles.

## 4. Concluding remarks

In this paper we have derived the exact solution to the flow inside and around a circular drop falling at low Reynolds number, with an arbitrarily varying interfacial tension. By avoiding the use of a surfactant advection-diffusion equation at the interface, we are able to obtain analytical solutions to the flow, which has not been possible in previous works. We demonstrate that when all the interfacial stress conditions are taken into account, one obtains a range of simultaneous solutions for the variation in interfacial tension and for the flow field, including exotic solutions such as the hovering drop. By appealing to conservation of energy, we restrict the allowed interval of solutions, and show that the terminal velocity of a falling drop must lie between the clean drop (Hadamard-Rybczynski) and the rigid sphere (Stokes) results.

To proceed with this approach, we propose a new model for how surfactants behave at the interface of a falling drop. Previous work has assumed the existence of a stagnant cap of surfactants on the top of a falling drop. In the present model we do not impose a specific surfactant distribution, but we introduce a simple model, called the continuous-interface model, which takes into account the force balance for surfactant molecules at the interface. It is demonstrated that the model gives a transition in terminal velocity as a function of drop radius that is consistent with experimental results. Moreover, by fitting the model to experimental results, we extract values for the critical radius as a function of bulk surfactant concentration. The model predicts that these should be related by a Langmuir isotherm, and indeed this is found to be true. We postulate that our model is more reasonable for fluid-like surfactant molecules, while the stagnant cap model may be appropriate for colloidal particles acting as surfactants. Future work may attempt to identify this difference experimentally.

Ending on a historical note, we have read with interest the recent paper by Hager (2012) about the life and work of Wilfrid Noel Bond, who, amongst other achievements, was the first person to observe and discuss the transition in terminal velocity that we aim to explain with our model. Bond's untimely demise was surely a great loss not only for his family, but also for the field of fluid mechanics research. †

The authors are grateful for stimulating discussions on these matters with Dr. Svend Tollak Munkejord and Professor Bernhard Müller. This work was funded by the project *Fundamental understanding of electrocoalescence in heavy crude oils* coordinated by SINTEF Energy Research. The authors acknowledge the support from the Petromaks programme of the Research Council of Norway (206976), Petrobras, Statoil and Wärtsilä Oil & Gas Systems.

## REFERENCES

Bond, W. N. 1927 LXXXII. Bubbles and drops and Stokes' law. *Phil. Mag.* **4** (24), 889–898.

Bond, W. N. & Newton, D. A. 1928 LXXXII. Bubbles and drops and Stokes' law. (Paper 2). *Phil. Mag.* **5** (30), 794–800.

Castiglione, P., Mazzino, A., Muratore-Ginanneschi, P. & Vulpiani, A. 1999 On strong anomalous diffusion. *Physica D: Nonlinear Phenomena* **134** (1), 75 – 93.

Chang, L. S. & Berg, J. C. 1985 The effect of interfacial tension gradients on the flow structure of single drops or bubbles translating in an electric field. *AIChE J.* **31** (4), 551–557.

Clift, R., Grace, J. R. & Weber, M. E. 1978 *Bubbles, drops and particles*. Academic Press.

† Those with an interest in history may also want to read the similar exposition on Wolfgang von Ohnesorge by McKinley & Renardy (2011).

Davis, R. E. & Acrivos, A. 1966 The influence of surfactants on the creeping motion of bubbles. *Chem. Eng. Sci.* **21** (8), 681 – 685.

Edge, R. M. & Grant, C. D. 1972 The motion of drops in water contaminated with a surface-active agent. *Chem. Eng. Sci.* **27** (9), 1709 – 1721.

Frumkin, A. & Bagotskaya, I. 1947 Unknown title (in Russian), cited in Levich: "Physicochemical Hydrodynamics". *Comptes Rendus de l'Académie des Sciences de l'URSS* **55** (131).

Frumkin, A. & Levich, V. 1947 On surfactants and interfacial motion. *Zh. Fiz. Khim* **21**, 1183–1204.

Giona, M., Cerbelli, S. & Vitacolonna, V. 2004 Universality and imaginary potentials in advection–diffusion equations in closed flows. *Journal of Fluid Mechanics* **513**, 221–237.

Griffith, R. 1962 The effect of surfactants on the terminal velocity of drops and bubbles. *Chem. Eng. Sci.* **17** (12), 1057 – 1070.

Hadamard, J. 1911 Mouvement permanent lent d'une sphère liquide et visqueuse dans un liquide visqueux. *Comp. Rend. Acad. Sci* **152** (25), 1735–1738.

Hager, W. H. 2012 Wilfrid Noel Bond and the Bond number. *J. Hydraul. Res.* **50** (1), 3–9.

Han, S.-I., Stapf, S. & Blümich, B. 2001 NMR imaging of falling water drops. *Physical Review Letters* **87** (14), 144501.

Hansen, E. B. 2005 Numerical simulation of droplet dynamics in the presence of an electric field. PhD thesis, NTNU.

Harper, J. F. 1973 On bubbles with small immobile absorbed films in liquids at low Reynolds numbers. *J. Fluid Mech.* **58** (3), 539–545.

Hatanaka, J., Maruta, K. & Asai, S. 1988 Terminal velocity of a contaminated drop at low Reynolds numbers. *Chem. Eng. J.* **39** (3), 185–189.

Kojima, M., Hinch, E. J. & Acrivos, A. 1984 The formation and expansion of a toroidal drop moving in a viscous fluid. *Phys. Fluids* **27** (1), 19–32.

Langmuir, I. 1917 The constitution and fundamental properties of solids and liquids. II. Liquids. *Journal of the American Chemical Society* **39** (9), 1848–1906.

Leal, L. G. 2007 *Advanced transport phenomena: Fluid mechanics and convective transport processes*. Cambridge University Press.

Lebedev, A. 1916 Stokes' law as applied to liquid balls (in Russian). *Zhurnal Fizicheskoi Khimii* **48**.

Levich, V. G. 1962 *Physicochemical hydrodynamics*, , vol. 689. Prentice-Hall Englewood Cliffs, NJ.

Lucassen-Reynders, E. 1996 Dynamic interfacial properties in emulsification. *Encyclopedia of Emulsion Technology* **4**, 63–91.

Masoud, H. & Stone, H. A. 2014 A reciprocal theorem for Marangoni propulsion. *Journal of Fluid Mechanics* **741**.

McKinley, G. H. & Renardy, M. 2011 Wolfgang von Ohnesorge. *Phys. Fluids* **23** (12).

Nordlund, I. 1913 The validity of Stokes' law for the motion of liquid drops in other liquids. *Arkiv för matematik, astronomi och fysik* **9** (13).

Reynolds, O. 1875 On the action of rain to calm the sea. *Nature* **11**, 279–280.

Rybzynski, W. 1911 Über die fortschreitende Bewegung einer flüssigen Kugel in einem zähen Medium. *Bull. Acad. Sci. Cracovie* .

Sadhal, S. & Johnson, R. E. 1983 Stokes flow past bubbles and drops partially coated with thin films. Part 1. Stagnant cap of surfactant film–exact solution. *J. Fluid Mech.* **126**, 237–250.

Savic, P. 1953 Circulation and distortion of liquid drops falling through a viscous medium. *Tech. Rep.* MT-22. National Research Council of Canada.

Silvey, O. W. 1916 The fall of mercury droplets in a viscous medium. *Physical Review* **7** (1), 106–111.

Stokes, G. G. 1851 On the effect of the internal friction of fluids on the motion of pendulums. *Camb. Phil. Soc. Trans.* **9**, 8.

Stone, H. A. 1994 Dynamics of drop deformation and breakup in viscous fluids. *Ann. Rev. Fluid Mech.* **26** (1), 65–102.

Taylor, T. D. & Acrivos, A. 1964 On the deformation and drag of a falling viscous drop at low Reynolds number. *J. Fluid Mech.* **18** (3), 466 – 476.

Thorsen, G., Stordalen, R. M. & Terjesen, S. G. 1968 On the terminal velocity of circulating and oscillating liquid drops. *Chem. Eng. Sci.* **23** (5), 413 – 426.

Volarovich, M. & Leont'yeva, A. 1939 Unknown title (in Russian), cited in Levich: "Physicochemical Hydrodynamics". *Acta Physiochimica URSS* **11**, 251.

Worthington, A. M. 1876 On the forms assumed by drops of liquids falling vertically on a horizontal plate. *Proc. R. Soc.* **25** (171-178), 261–272.

Young, N. O., Goldstein, J. S. & Block, M. J. 1959 The motion of bubbles in a vertical temperature gradient. *Journal of Fluid Mechanics* **6**, 350–356.

# Paper 6

# raaSAFT: a framework enabling coarse-grained molecular dynamics simulations based on the SAFT-$\gamma$ Mie force field

Åsmund Ervik[a,*], Guadalupe Jiménez Serratos[b], Erich A. Müller[b]

[a] *NTNU, Dept. of Energy and Process Engineering, Kolbjørn Hejes v 1B, NO-7491 Trondheim, Norway*
[b] *Imperial College London, Dept. of Chemical Engineering, South Kensington Campus, London SW7 2AZ, United Kingdom*

**Abstract**

We describe here raaSAFT, a Python code that enables the setup and running of coarse-grained molecular dynamics simulations in a systematic and efficient manner. The code is built on top of the popular HOOMD-blue code, and as such harnesses the computational power of GPUs. The methodology makes use of the SAFT-$\gamma$ Mie force field, so the resulting coarse grained pair potentials are both closely linked to and consistent with the macroscopic thermodynamic properties of the simulated fluid. In raaSAFT both homonuclear and heteronuclear models are implemented for a wide range of compounds spanning from linear alkanes, to more complicated fluids such as water and alcohols, all the way up to nonionic surfactants and models of asphaltenes and resins. Adding new compounds as well as new features is made straightforward by the modularity of the code. To demonstrate the ease-of-use of raaSAFT, we give a detailed walkthrough of how to simulate liquid-liquid equilibrium of a hydrocarbon with water. We describe in detail how both homonuclear and heteronuclear compounds are implemented. To demonstrate the performance and versatility of raaSAFT, we simulate a large polymer-solvent mixture with 300 polystyrene molecules dissolved in 42 700 molecules of heptane, reproducing the experimentally observed temperature-dependent solubility of polystyrene. For this case we obtain a speedup of more than three orders of magnitude as compared to atomistically-detailed simulations.

*Keywords:* Molecular Dynamics, GPU, HOOMD-blue, coarse-grained, SAFT-$\gamma$ Mie

**PROGRAM SUMMARY**

*Manuscript Title:* raaSAFT: a framework for enabling coarse-grained molecular dynamics simulations based on the SAFT-$\gamma$ Mie force field
*Authors:* Åsmund Ervik, Guadalupe Jiménez Serratos, Erich A. Müller
*Program Title:* raaSAFT
*Journal Reference:*
*Catalogue identifier:*

*Corresponding author
*Email addresses:* asmunder@pvv.org (Åsmund Ervik), m.jimenez-serratos@imperial.ac.uk (Guadalupe Jiménez Serratos), e.muller@imperial.ac.uk (Erich A. Müller)

*Licensing provisions:* MIT License
*Programming language:* Python
*Computer:* Any computer, optionally with Nvidia GPU(s)
*Operating system:* Linux, Mac OSX
*RAM:* Depends on number of atoms and cutoff size
*Number of processors used:* Any number of CPUs or GPUs
*Supplementary material:*
The code is hosted on http://bitbucket.org/asmunder/raasaft
*Keywords:*
Molecular Dynamics, GPU, HOOMD-blue, GROMACS, coarse-grained, SAFT-$\gamma$ Mie
*Classification:*
7.7 Other Condensed Matter inc. Simulation of Liquids and Solids, 16.13 Condensed-phase simulations
*External routines/libraries:* HOOMD-blue [1][2]
*Nature of problem:*
The behaviour and properties of simple and complex fluids, including mixtures
*Solution method:*
Coarse-grained molecular dynamics using the SAFT-$\gamma$ Mie force field [3].
*Restrictions:*
Ions and ionic compounds are not supported yet. Jobscripts running with Python 2 require HOOMD-blue v1.3 or newer; for Python 3 there is no such restriction.
*Unusual features:*
Uses object-oriented programming to make reuse and sharing of models very simple. Allows the simulation to be set up and executed completely programatically, i.e. without the use of a GUI or preprocessor. Force field parameters are available from an online database with more than 6000 molecules, `http://www.bottledsaft.org` [4].
*Running time:*
On a single high-end GPU in 2015 (Nvidia K40), around 2.5 nanoseconds per hour of walltime for a million atoms (not counting hydrogens).

## References

[1] J. A. Anderson, C. D. Lorenz, A. Travesset, General purpose molecular dynamics simulations fully implemented on graphics processing units, Journal of Computational Physics 227 (2008) 5342 – 5359.
[2] J. Glaser, T. D. Nguyen, J. A. Anderson, P. Lui, F. Spiga, J. A. Millan, D. C. Morse, S. C. Glotzer, Strong scaling of general-purpose molecular dynamics simulations on GPUs, Computer Physics Communications 192 (2015) 97 – 107.
[3] E. A. Müller, G. Jackson, Force-field parameters from the SAFT-$\gamma$ equation of state for use in coarse-grained molecular simulations, Annual review of chemical and biomolecular engineering 5 (2014) 405–427.
[4] Å. Ervik, A. Mejía, E. A. Müller, Bottled SAFT: a web app providing SAFT-$\gamma$ Mie force field parameters for thousands of molecular fluids, In preparation. (2016).

## 1. Introduction

The coarse-grained molecular dynamics (MD) simulation of simple and/or complex fluids is a popular method for gaining physical insight into complex phenomena that elude investigations by experiments or by theory. By coarse-graining we mean here the procedure of lumping together the effect of a few individual atoms into a single sphere or

*bead* that forms the basic element of the MD simulation. These beads interact with each other through an effective spherically symmetric pair potential. A popular variant is the Lennard-Jones or (12,6) potential, consisting of a repulsive term proportional to $r^{-12}$ and an attractive term proportional to $r^{-6}$, where $r$ is the distance between two beads. If the beads are to interact with beads of another type, a cross-interaction potential must also be specified in some way.

As is natural for a popular technique, there exists in the literature a large number of procedures for coarse-graining the description of molecular fluids. Most of these procedures are of the so called bottom-up type, meaning that one starts with an atomistic representation of some molecule, and then tries to build representative beads typically containing 2-4 atoms (not counting hydrogen) which represent, in some integrated fashion, this atomistic structure. This procedure is often carried out via a trial-and-error procedure. The resulting potentials describing the interaction between two beads are not coupled to the macroscopic properties of the fluid, and typically lack both transferability and robustness. A recent review of these techniques is given in [1].

As alternatives to this we have the top-down, or thermodynamic, coarse-graining procedures. A popular example is the MARTINI force field [2], originally designed for biomolecular applications such as simulation of lipids. In MARTINI, the super-atoms (i.e. beads) are parametrised by matching the free energy of transfer between water and organic solvents. The individual beads may then be connected in a group-contribution fashion to model larger molecules. The model is based on the Lennard-Jones potential and reports a finite set of parameters "building blocks"; both aspects somehow limit the applicability.

The approach used here, namely the SAFT-$\gamma$ Mie force field parametrization [3], is another form of top-down coarse-graining procedure. In this approach, the interaction between a pair of beads is through the Mie potential, also called the generalised Lennard-Jones potential, seen in Equation (1) below. In the SAFT-$\gamma$ Mie approach, the four parameters of this potential have a one-to-one correspondence with the parameters of the SAFT-$\gamma$ equation of state (SAFT stands for Statistical Associating Fluid Theory). By fitting this equation of state to experimentally measured macroscopic thermodynamic properties, e.g. the densities of the saturated liquid and vapour pressure states of some compound as a function of temperature, one fixes also the parameters of the Mie potential. The crucial point of the approach is that fluid properties obtained from MD simulations using this potential agree very well both with the equation of state and with the original experimental data. Furthermore, the molecular simulations can accurately predict properties that were not used in the initial fit, e.g. the interfacial tension, heat capacity or the speed of sound of the fluid in question. The accuracy has been found to be comparable with atomistically detailed models that have runtimes two orders of magnitude larger [4].

The Mie potential is given as a function of the distance $r$ between two beads,

$$\begin{aligned} V(r) =& C(n,m)\,\epsilon\left[\left(\frac{\sigma}{r}\right)^m - \left(\frac{\sigma}{r}\right)^n\right], \\ C(n,m) =& \left(\frac{m}{m-n}\right)\left(\frac{m}{n}\right)^{n/(m-n)}. \end{aligned} \tag{1}$$

The potential consists of a repulsive ($r^{-m}$) and an attractive ($-r^{-n}$) term, i.e. a pair

of beads experience repulsion when they are very close, attraction when they are further apart, and no force at infinite separation. The potential has a single global minimum, often called the energy well, defined at a value of $\epsilon$. The parameter $\sigma$ specifies the centre-to-centre distance at which the potential switches from a repulsive to an attractive interaction, hence it is often taken to be the effective diameter of the bead for visualisation purposes. The two exponents $m, n$ of the attractive and repulsive terms can be freely adjusted, allowing for a more complex balance between attractive and repulsive forces than the standard Lennard-Jones potential ($m = 12, n = 6$). We remark that compared to the Lennard-Jones potential, the number of free parameters for the Mie potential increases not by two, but by one, since two of the four parameters in Equation (1) are intimately related [5].

Taking this as our fundamental approach, we have constructed a Python framework that makes it straightforward to set up and run MD simulations using coarse-grained models with the SAFT-$\gamma$ Mie approach. The framework is called raaSAFT (pronounced "raw saft") which in Scandinavian languages means "pure fruit juice"; "raa" may also be translated into "coarse". The framework leverages existing MD codes to perform the actual simulations. HOOMD-blue [6, 7], a modern "GPU-first" code, is the option most closely integrated in raaSAFT. One may also use GROMACS [8], a popular and highly capable MD code, which is however more loosely integrated.

The speedup obtained by leveraging the potential of graphics processing hardware to perform MD simulations is particularly impressive [9], and rivals the speedup obtained by simplifying the molecular model (i.e. coarse-graining). The combination of both approaches is guaranteed to allow the exploration of both long time scales and large system sizes which are required to study a wealth of interesting and complex fluid behaviour from a molecular viewpoint. As an example, the polymer simulation described in Section 4.2 would have a runtime of the order of 10 years if one were to use an atomistically-detailed model and a four-CPU workstation. On a four-GPU workstation with the coarse-grained model, the runtime is 135 hours.

A force field for coarse-grained MD simulation is nothing without force field parameters which represent different molecules that can be simulated. A number of publications have appeared which detail the parametrization procedures to obtain SAFT-$\gamma$ Mie force field parameters for carbon dioxide [10], greenhouse gases and refrigerants [11], benzene [12], water [13], mixtures of water, carbon dioxide and n-alkanes [14] etc. An alternative approach to fitting the equation of state for each and every compound is the use of the "M&M" correlation developed by Mejía et al. [15]. This correlation computes the force field parameters directly from the critical properties of the molecular fluid in question. These properties are readily available and tabulated for a vast number of fluids.

Building on the M&M correlation, we have constructed a web application called Bottled SAFT [16], providing force field parameters for over 6000 compounds. It is available at `http://www.bottledsaft.org` and is free and open source. The user may search this database by name, chemical formula or CAS number. Once a result is obtained, the webpage produces a script which implements the result directly in raaSAFT.

The outline of this paper is as follows. We first present a description of raaSAFT, including the physical units employed, how to install raaSAFT, and how to set up and run simulations. As an example we consider a simulation of liquid-liquid equilibrium of

n-octane and water, and illustrate how to process the results of this simulation. We give a brief overview of how to use the GROMACS backend. We then proceed to describe how different compounds are implemented, first for the simple homonuclear case and subsequently for a full heteronuclear example. User-defined compounds are implemented in the same way. Next, we present results showcasing the applications and performance of raaSAFT, including weak and strong scaling with various hardwares and systems. Finally, we illustrate the versatility and performance of raaSAFT with a large simulation of polystyrene in a solution of heptane, where we reproduce the temperature-dependent solubility found in experiments.

## 2. Program description

## 3. Software design and use

### *3.1. Unit conventions*

While some codes (e.g. GROMACS) establish a unit convention, others employ "natural units" that leave the user free to choose the physical units, e.g. the units in which the energy is measured. HOOMD-blue falls in the latter category. In this section we specify the units chosen for raaSAFT. When running with GROMACS as a backend, these units are converted into the GROMACS unit convention.

The fundamental units in HOOMD-blue are length, energy and mass. Furthermore, temperature is given as the thermal energy $k_{\mathrm{B}}T$. The choice of units for raaSAFT is given in Table 1. With this, the units of derived quantities are as given in Table 2, in particular the time unit is 1 ps (picosecond). The energy in the Mie potential is specified as $\epsilon/k_{\mathrm{B}}$ in units of K, where $k_B = 8.3144622$ (J/mol)/K is the Boltzmann constant, as is the convention in papers using the SAFT-$\gamma$ Mie force field.

Table 1: Fundamental units used by raaSAFT

| Quantity | Symbol | Unit |
|---|---|---|
| distance | r | 1 Å $= 10^{-10}$m |
| energy | $\xi$ | 10 J/mol |
| mass | m | 1 g/mol |

Table 2: Derived units used by raaSAFT. Here $N_{AV}$ is Avogadro's number, $\epsilon_0$ is the vacuum permittivity, and statC is the stat-Coloumb unit of charge, equivalent to $1/(10c)$ Coulomb where $c$ is the numerical value of the speed of light in m/s.

| Quantity | Symbol | Formula | Unit |
|---|---|---|---|
| time | t | $\sqrt{mr^2/\xi}$ | 1 ps |
| velocity | v | $r/t$ | 1 Å/ps = 100 m/s |
| temperature | T | $k_{\mathrm{B}}T_{\mathrm{Kelvin}}$ | 10 J/mol |
| pressure | P | $\xi/r^3$ | $10^{30}/N_{AV}$ Pa |
| charge | q | $\sqrt{4\pi\epsilon_0 r\xi}$ | 1 statC$/\sqrt{N_{AV}} = 1/(2997924580\sqrt{N_{AV}})$ C |

*3.2. Installing raaSAFT*

Since raaSAFT is purely a Python package, it can be installed easily on any computer, even if the user is not an administrator (e.g. on a cluster). The first point is to ensure that HOOMD-blue is installed. Then raaSAFT can be installed from the PyPi package repository using the command `pip install raasaft`. The package may be upgraded to the latest version with the command `pip install -upgrade raasaft`.

For advanced usage, the code may instead be installed by cloning the git repository at `http://bitbucket.org/asmunder/raasaft` (code provided under the MIT license). Contributions, either with implementations of new models or improvements to the framework, are very welcome.

When raaSAFT has been installed with `pip`, it is advisable to create a directory for running raaSAFT simulations, called e.g. `raasaft`. Jobscripts for different simulations are best organised by putting them in different subdirectories of this directory. It is also advisable to run the command `raasaft_init` in the `raasaft` directory. This command populates the directory with the `tutorials`, `replication` and `mysaft` folders. As the name implies, `tutorials` contains example jobscripts that demonstrate how to use raaSAFT. These may be run e.g. as `hoomd ex1.hoomd`. Implementations of user-defined compounds are put in the `mysaft` folder; this is described in more detail at the end of Section 3.5.1.

The `replication` folder contains jobscripts from publications using raaSAFT. Authors who use raaSAFT for their simulations are invited to submit their jobscripts along with a brief README file to this repository, either as a pull request on Bitbucket or by email to one of us. To seed this folder we have included jobscripts for the present work, and will be adding more in the near future.

*3.3. The interface for running a simulation*

The user interface for raaSAFT is very similar to the user interface for HOOMD-blue [6, 7], which raaSAFT was originally built around. This interface consists of a jobscript written in Python, in which one imports the required libraries, specifies the desired system components, the number of unique components, as well as the initial density, and then initialises the system. Currently, this system initialisation is done with the `create_random_polymers()` function from HOOMD-blue, at a low density (even for liquid simulations), since this random initialisation is not very sophisticated. The desired density is achieved by simulating initially in the isobaric-isothermal (NPT) ensemble for a short time. Future versions of raaSAFT may include a more sophisticated system initialisation.

In this section we will give a thorough walkthrough of a typical jobscript for running a simulation with raaSAFT. The jobscript in its entirety can be found in Appendix A. The first task is to import `hoomd_script` as well as the desired components of raaSAFT. After this, a call is made to the function `context.initialize()` which was introduced as mandatory in recent versions of HOOMD-blue. To avoid having to worry about whether to call this function or not, raaSAFT "monkey patches" it such that it is available (and does nothing) in case the simulation is using an earlier HOOMD-blue version where the function is not defined. These first lines are shown in Listing 1.

Following this, the components to simulate are selected. In the simplest case this is just a single compound, e.g. for simulating vapour-liquid equilibrium. In more

```
#!/usr/bin/python
from hoomd_script import *
from raasaft.main import *
from raasaft.alkanes import *
from raasaft.water import *

context.initialize()
```

Listing 1: Importing HOOMD-blue, raaSAFT, and initialising the context.

complicated cases it may involve a large number of homonuclear and heteronuclear components, e.g. for simulating a crude oil. Each component constructor takes the argument `count` which specifies how many molecules of this component to include. Some constructors take additional arguments, e.g. alkane models require the carbon number of the desired alkane. After selecting the components, the simulation box is set up and the system is then initialised. An example of this is shown in Listing 2, where we specify the components n-octane and water, using the force field parameters from [14] and [13], respectively.

```
C8 = HomoAlkane(C=8, count=1e4)
H2O = BioWater(count=6e4)
components = [C8,H2O]

theBox = setupSimBox(components, elong=3.0, packing=0.1)
system = setupSimulation(components, theBox)
```

Listing 2: Setting the desired components, creating the box, and initialising the system with randomly placed beads.

At this point the data structures containing all the beads have been set up. The standard HOOMD-blue initialisation sets up the initial positions and velocities for all the beads, and also adds the bonds between beads where this has been specified. To proceed one must inform HOOMD-blue about the potential for each components, and the cross-interactions between components. The Lafitte combination rules [17] are used for the cross-interactions, where one free parameter $k_{ij}$ is available for tuning. If unspecified, the $k_{ij}$ is set to zero by raaSAFT, which is a reasonable starting point. The tuning should be done by comparing to experimental values of some macroscopic property, for instance interfacial tension, solubilities, vapour-liquid compositions etc.

To implement this in the jobscript, an important variable to define is the cutoff to be used. It is either set automatically to a conservative default for the selected components, namely $6\sigma$ for the largest $\sigma$ in the system, or it is specified by the user. The long cutoff values ($> 5\sigma$) typically provide a more accurate fit to the SAFT-$\gamma$ equation of state predictions, but incur a speed penalty as more force field evaluations are required per time step. Very short cutoffs (e.g. $2.5\sigma$) provide the fastest runs but compromise on the quantitative aspects of the result.

Having specified the cutoff, one then specifies any nonzero $k_{ij}$ to be used in the cross-interactions. Next, a HOOMD-blue function is used to create the `table` object holding the potentials, and one then sets the potential parameters and all the cross

interactions. This is shown in Listing 3. Note in particular the simplicity of setting the cross-interactions for all the $N$ different bead types. This single line sets all $N^2/2$ cross interactions. Having $N = 20$ is not unreasonable for a complicated system with several heteronuclear compounds, say a model for crude oil. If the jobscript were to have 200 lines of code just for setting the cross-interactions, it would be hard to read and easy to make mistakes. The current code uses the Lafitte combination rules and the `combinations()` function from the `itertools` module in the Python standard library to dynamically set all cross-interactions. Setting the cutoff, cross-interactions and tabulating the potentials is shown in Listing 3.

```
setCutoff(components, cutoff=20)
setCrossInteraction(C8, H2O, kij=0.3205)
table = pair.table(width=1000)
setPotentialCoeffs(components, table)
setAllCrossCoeff(components, table)
```

Listing 3: Setting the cutoff, potentials and cross interactions for all components. The cross interactions use the Lafitte combination rules [12] with a default $k_{ij} = 0$, other values may be set as shown here.

The next item on the agenda is to specify the forces which enact the bonds. With raaSAFT, the equilibrium length and constant for each bond is contained in the implementation of each compound, so the job script just has to call a function to set these in the data structures used by HOOMD-blue. Typically harmonic bonds are used, with bond constants either calculated to match experimental evidence or atomistic simulations (see e.g. the bond constants for alkane chains in [14]), or bond constants are set to a large numerical value in order to approximate rigid bonds. The bonds are set with a single command in Listing 4.

For compounds consisting of three or more beads, one is often also interested in constraining some of the angles between sets of three beads. Here again harmonic potentials are used, and there is again a distinction between "realistic" values of the angle potential parameters, and numerical values intended to enforce a specific angle, such as the cis angle for an alkyl group. A difference from the bonds is that angles are not added to the system by HOOMD-blue in the system initialisation. Therefore raaSAFT must add the appropriate angles for each component. The bonds are added and set with a single command in Listing 4.

```
setupBonds(system, components)
setupAngles(system, components)
```

Listing 4: Setting up the potentials that govern the bonds and angles.

When this has been completed, the remaining functions to be called from the jobscript are the same as in a standard HOOMD-blue jobscript. For completeness we shall describe the typical usage of these functions in a raaSAFT-based jobscript. All functions mentioned in the following are thus from HOOMD-blue. The first are two boilerplate lines which are used in essentially all job scripts. These are shown in Listing 5. The first line creates a group containing all beads, and the second sets the time step and

integration mode. This mode is the standard one, unless one wants to perform an energy minimisation using the FIRE algorithm [18].

```
all = group.all()
mode = integrate.mode_standard(dt=0.001)
```

Listing 5: Setting the integration mode and timestep.

As mentioned in the first paragraph of this section, the system is generally initialised at a low density. This has two purposes: first, it avoids blowup during the first few hundred time steps by reducing the number of beads placed very close by the random initialisation. The second is that this largely avoids the tangling of complicated molecules. To further reduce the risk of blowup, the first few hundred time steps can be performed in the NVE (or microcanonical) ensemble with a limiter, i.e. restricting how far a molecule is allowed to move in a timestep no matter how large the forces on it are. It is then beneficial to run the system in the NVT (or canonical) ensemble for a few thousand time steps before increasing the system density. The purpose of this short NVT run is to randomise and thermally equilibrate the configuration in preparation for the subsequent resizing of the system. This resizing is done by two different methods, depending on whether a vapour-liquid equilibrium (VLE) or a fully liquid system is desired. For a VLE, it is simpler to just resize the box manually, such that the resulting overall density is between that of the liquid and of the vapour. Upon equilibration this will produce a slab of liquid surrounded by its coexisting saturated vapour. For a fully liquid system, one has to run in the NPT (or isothermal-isobaric) ensemble to obtain the desired pressure for the liquid, since the pressure is highly sensitive to the box size for an essentially incompressible liquid and thus manual tuning is unlikely to succeed. In Listing 6, we show how the system is run in the different ensembles to obtain a fully liquid system in the end.

```
relaxation = integrate.nve(group=all, limit=0.01)
run(1e3)
relaxation.disable()

Temp = 298.15*kBby10
nvt = integrate.nvt(group=all, T=Temp, tau=0.5)
run(4e4)
nvt.disable()

npt = integrate.npt(group=all, T=Temp, P=1*ConvFromBar, tau=0.5, tauP=0.5)
run(1e5)
```

Listing 6: Integrating the system to obtain a fully liquid system.

When simulating a liquid-liquid or vapour-liquid equilibrium, it is convenient to use a simulation box which is elongated in one direction, since this causes the formation of well-defined interfaces normal to the elongated direction. This setup will typically produce two slabs of segregated fluids withing the box, which can be considered as a tiny patch of some macroscopic fluid interface due to the periodic boundary conditions; from this we can compute e.g. the interfacial tension. A system of immiscible liquids

simulated using the SAFT-$\gamma$ Mie force field will spontaneously phase separate even when started from random initial conditions, but it may initially form clusters and/or multiple slabs of liquid, which then take a very long time to coalesce into two slabs. This is due to the inherent limitations of the timescales in simulation being small as compared to the characteristic timescales for cluster diffusion.

A useful way of speeding this up consists in applying an external potential, varying in the direction normal to the desired liquid-liquid interface, taking the form of a periodic hyperbolic tangent function with opposite magnitudes for the two liquids. It is important that the width of this external potential is large enough that a substantial number of particles experience a force from the gradient of the potential. After the periodic potential has been applied for some time, it is turned off. The system's total momentum is then set to zero, as it may have gained an undesirable net momentum due to random initial alignment of the two liquids with the potential. Running with an external periodic potential and subsequently removing the net momentum is shown in Listing 7.

```
periodic = external.periodic()
periodic.force_coeff.set(C8.Name, A = 400.0, i=2, w=0.2, p=1)
periodic.force_coeff.set(H2O.Name, A = -400.0, i=2, w=0.2, p=1)
run(1e6)
periodic.disable()
update.zero_momentum()
npt.disable()
```

Listing 7: Speeding up liquid-liquid separation with an external potential.

After having resized the system, and optionally having obtained a system with two slabs of liquid, one is ready to do a production simulation. It is then desirable to log integral quantities such as the temperature, energy etc., and to save the individual trajectories of the beads. Both of these operations are done using standard HOOMD-blue functionality, as shown in Listing 8.

```
xml = dump.xml(filename='con.xml', vis=True)
dump.dcd(filename='dump.dcd', period=1e4)
logQuant = ['time', 'pair_table_energy', 'pressure', 'temperature',
            'pressure_xx', 'pressure_yy', 'pressure_zz']
analyze.log(filename='log.dat', quantities=logQuant, period=1e2, header_prefix='')
```

Listing 8: Specifying the logging of scalar quantities and the dumping of trajectories for visualisation.

To do a production run, i.e. a simulation which we can analyse to get the information we seek, we switch from the NPT to the NVT ensemble, and run for some time. At this point, we may increase the time step size. At the end of the production run it is useful to save a restart file, such that one can continue the simulation from that point. This is shown in Listing 9.

The entire jobscript (see Appendix A), contained e.g. in the file `octane-water.py`, may now be executed with the command `hoomd octane-water.py`. After the simulation has finished (or while it is running) we may analyse the the results. This typically consists

```
mode.set_params(dt=0.01)
nvt = integrate.nvt(group=all, T=Temp, tau=0.5)
run(2e6)
xml = dump.xml(filename='restart.xml', vis=True, velocity=True)
```

Listing 9: Doing the production run.

in either plotting some scalar quantity from the log file, or opening the .dcd and .xml files in VMD [19] and analysing the results there.

For our liquid-liquid simulation we may plot the interfacial tension $\gamma$, computed from the diagonal elements of the pressure tensor as

$$\gamma = \frac{L_z}{2} \int \left( P_{zz} - \frac{1}{2}(P_{xx} + P_{yy}) \right) dz. \tag{2}$$

These values are sampled every 100 time steps in the simulation. In Figure 1 we plot the cumulative average (green line) and a rolling average over 200 samples (blue dots) together with the experimental value of 51 mN/m (dashed orange line). The agreement is very good. The rolling average gives an impression of the uncertainty, but it would be better to use a quantitative measure, e.g. the method discussed in the appendix of [20]. This will be considered in future versions of raaSAFT.

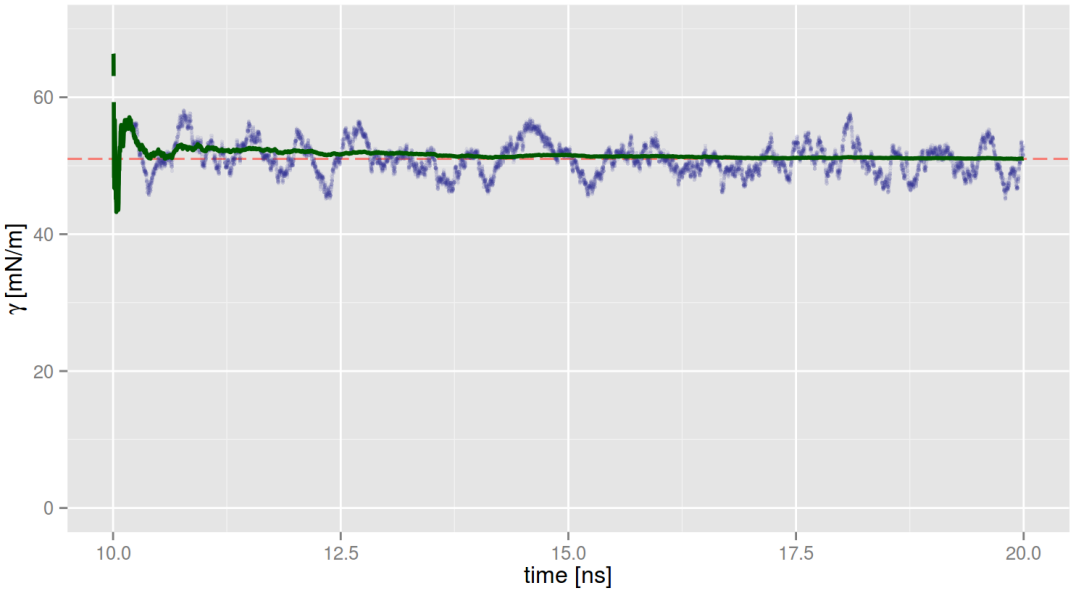



Figure 1: The interfacial tension of octane-water at 298 K and 1 bar, computed using Equation (2), shown here for the last half (10 ns) of the production run. Values are logged every 100 time steps (0.01 ns). Green line: cumulative average. Blue points: rolling average over 200 values. Dashed orange line: experimental value from [21].

We may also use VMD to gain a better insight into the system behaviour. Of particular use is the density profile plugin [22] to plot the density variations in the

z-direction, showing the densities of each phase and also the solubilities of these two phases in each other. For octane and water, which are immiscible fluids, this is not very interesting, but for other systems it can provide valuable insights; see [23] for a comprehensive overview.

### *3.4. Using GROMACS as the backend*

As mentioned, raaSAFT additionally supports running simulations with GROMACS. In this case, the first part of the jobscript looks identical to that described in the previous section, up to and including Listing 6. At this point, the system is in the desired state and the positions and velocities of the beads are well-behaved such that GROMACS can easily restart from this point. Then, instead of proceeding with the simulation using HOOMD-blue, the user tells raaSAFT to write the system configuration to the files which GROMACS uses for running the simulation, with a single function: `gmxWriteSystemFiles(system,components)`.

This function goes through the system and the components and

- Converts from the units described in Section 3.1 to GROMACS units
- Constructs the .itp file that specifies the compounds, their bonds, angles etc.
- Constructs the .mdp file with default parameters for the job, including the interactions and cross-interactions to include
- Constructs the system.top file that sets the number of each compound
- Tabulates the potential for each interaction and cross-interaction into .xvg files
- Constructs the .gro file that contains the initial position and velocity of each bead
- Finally, prints to screen the commands the user should invoke to run the simulation

Presently, the generation of these files, in particular the initial position and velocities of each bead, requires that HOOMD-blue is installed (it may be compiled and used without GPU support if GPUs are not available). This restriction may be removed in future versions of raaSAFT.

### *3.5. Implementing a compound*

In raaSAFT, the definition of a given compound, be it homo- or heteronuclear, is abstracted away from the jobscript. There are two important reasons for this choice. One, it allows the efficient reuse of a compound implementation in many jobscripts. Two, it separates the compound implementation from the simulation setup, which are two logically distinct concepts.

In the following subsections we describe how compounds are implemented in the raaSAFT code. We first discuss the simpler case of homonuclear compounds, and then extend the discussion to heteronuclear compounds, which use several homonuclear compound implementations as part of their implementation. Finally, we discuss Bottled SAFT, our online library of compounds which users may search and download compound implementations from, as well as some implementation details that may be of interest to others.

*3.5.1. Homonuclear compound*

The implementation of a homonuclear compound in raaSAFT is straightforward. Starting with the simplest case of a one-bead model, the implementation only has to specify the parameters of the Mie potential for this compound. Taking as an example the `Benzene1` compound (from [12]), which is a single-bead model for benzene. It is implemented as shown in Listing 10. Here we see that the compound extends the base

```
class Benzene1(MieCG):
    def __init__(self,count):
        self.Name = "Benzene"
        self.Epsilon = 658.17*kBby10
        self.Sigma = 5.293
        self.N = 32
        self.M = 6
        self.Segments = 1
        self.Mass = 78.11 / self.Segments
        self.Citation = "DOI: 10.1080/00268976.2012.662303"
        MieCG.__init__(self,count)
```

Listing 10: Specification of the one-bead benzene model

`MieCG` class. We specify the name, parameters for the Mie potential, number of segments, molar mass, and a reference to the literature for this set of parameters. Note that the number given for `Epsilon` is the energy divided by the Boltzmann constant $k_B$, in K, as given in [12]. This is multiplied by $k_B/10$, giving an energy in the expected units (10 J/mol). The base class constructor then sets properties common to all compounds based on the SAFT-$\gamma$ Mie force field, and contains convenience functions which are useful for all such compounds.

A more complicated example is a homonuclear compound with more than one bead and with some internal structure. As an example, we may take the three-bead model for benzene implemented in `Benzene3` (also from [12]), shown in Listing 11. As for the

```
class Benzene3(MieCG):
    def __init__(self,count):
        self.Name = "Benzene"
        self.Epsilon=258.28*kBby10
        self.Sigma=3.490
        self.N=11.58
        self.M=6
        self.Segments = 3
        self.Mass = 78.11 / self.Segments
        self.Citation = "DOI: 10.1080/00268976.2012.662303"
        self.BondConstant = RigidBondConstant
        self.Bonds = [(0,1), (1,2), (2,0)]
        MieCG.__init__(self,count)
```

Listing 11: Specification of the three-bead benzene model

one-bead model, we specify the name, Mie potential parameters, number of segments and citation. In addition we must specify the bond constant and the bond layout. Note

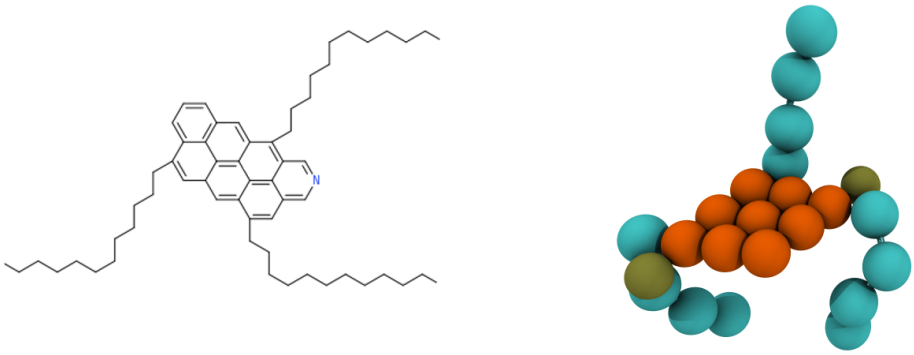



Figure 2: Molecular structure (left) and coarse grained version (right) of the model asphaltene molecule.

that the beads are zero-indexed. The bond layout here specifies that there are bonds between beads 0 and 1, 1 and 2, 2 and 0; this gives a triangle. The bond length is set automatically to give tangential bonding.

Before proceeding to the heteronuclear case, we will explain the purpose of the previously mentioned `mysaft` folder which is among those created by the `raasaft_init` command. This folder is used for implementing user-defined compounds. An example of a simple homonuclear compound is included in the folder. The `mysaft` folder is structured as a Python package, so the user should put the implementation of the desired compound, e.g. as in Listing 10 and Listing 11, in one or more files in this folder. Say these are put in the file `benzene.py`. The user must then append the `mysaft` folder path to the `$PYTHONPATH` environment variable, e.g. by running the command `export PYTHONPATH=$PYTHONPATH:$PWD` from inside the folder. The compounds implemented in `benzene.py` can then be used in a jobscript by putting `from mysaft.benzene import *` at the top of the jobscript.

*3.5.2. Heteronuclear compound*

The implementation of a heteronuclear compound is naturally more involved, because these models are inherently more complicated. As an example we consider a model coarse-grained asphaltene molecule, which has been successfully used in simulations of real crude oils [24]. The coarse-grained version and the molecular structure are shown in Figure 2. Asphaltenes are a major source of disruption in the production and processing of petroleum. Due to their tendency to foul pipelines up to the point of blockage, they have been called "the cholesterol of crude oil". Despite more than a century of scientific research, there are many aspects of asphaltenes that elude our understanding. For recent reviews see e.g. [25, 26, 27, 28]. Coarse-grained molecular dynamics simulation is a very promising tool for increasing our understanding of asphaltenes.

Here we show three listings with the implementation of this compound. The first shows the general setup, the second shows the bond setup, and the third shows the angle setup. In the first, Listing 12, we begin by setting the different types of beads making up the molecule, here three. We put these into a list for later convenience. We then give the compound a name and a short name; the latter is handy for the bond and angle data structures. We set the number of segments and specify the number and sequence of the different bead types. Finally we call the parent class constructor for `MieCGHet`.

```
class APCnew(MieCGHet):

    def __init__(self,count):
        # Initialize the different bead types
        self.C12 = MnMAlkane(C=12)
        self.ANT = Anthracene()
        self.PY  = Pyridine()
        # Put these in a list
        self.Components=[self.C12, self.ANT, self.PY]
        # Give this compound a name and a short name
        self.Name = "AsphalteneContinental"
        self.ShortName = "APC"
        # Set the number of segments
        self.Segments = 23
        # This sets the particle numbering: (0,1,2,3) are alkanes, (4,5,6) are
        # anthracenes, etc.
        self.Def = [self.C12]*4 + [self.ANT]*3 + [self.PY] + [self.ANT]*3 + \
                   [self.PY] + [self.ANT]*3 + [self.C12]*8
        # Initialize the base class instance
        MieCGHet.__init__(self,count)
```

Listing 12: Specification of a coarse-grained asphaltene model: general

Once the general specification is completed, we proceed to the bonds. For this model we need four different types of bonds, since there are four different bond lengths. The canonical names of these four bonds are then constructed from `self.ShortName`. The majority of this listing is taken up by the specification of `self.Bonds` which stores information about which beads are bonded together and by what type of bonds. Finally, `createBondSpec()` creates additional data structures based on `self.Bonds` and the list of bond constants given as an argument to `createBondSpec()` (here the same constant for all).

Once the bonds are specified, we proceed to the angles. The present model has two types of angles: one that constrains the aromatic core to form a flat structure, and one that allows some flexibility in the alkane tails. Again, the names of the angles are constructed, and `self.Angles` takes up the majority of this listing. In this list, e.g. the tuple `(0, 1, 2, at1)` specifies that the angle between the `(0,1)` and the `(1,2)` bonds is to be of type `at1`. Finally, additional data structures are created based on this list and on the angle constants and equilibrium angles given as arguments to `createAngleSpec()`. Here the first is given as a flexible angle suitable for linear alkanes [14], while the second is given as a more rigid angle.

### *3.6. Bottled SAFT*

As mentioned, we have built a large database of force field parameters based on the M&M correlation presented in [15]. This database can be searched with the web application Bottled SAFT (`http://www.bottledsaft.org`). It contains representations of more than 6000 molecular fluids, and users may search by name, CAS number, or chemical formula. Once a result is found, the user is presented with a table of force

```
# Bond type 1 is alkane-alkane, type 2 is alkane-anthracene,
# type 3 is anthracene-anthracene, type 4 is anthracene-pyridine.
self.NrBondTypes = 4
bt1, bt2, bt3, bt4 = self.getBondNames()
# Now configure the bonds
self.Bonds =[( 0, 1,bt1), ( 1, 2,bt1), ( 2, 3,bt1),
             ( 3, 6,bt2), ( 4, 5,bt3), ( 4, 8,bt3),
             ( 4, 9,bt3), ( 5, 6,bt3), ( 5, 9,bt3),
             ( 5,10,bt3), ( 6, 7,bt4), ( 6,10,bt3),
             ( 8, 9,bt3), ( 8,12,bt3), ( 8,13,bt3),
             ( 9,10,bt3), ( 9,13,bt3), ( 9,14,bt3),
             (10,14,bt3), (11,12,bt4), (12,13,bt3),
             (12,15,bt2), (13,14,bt3), (14,19,bt2),
             (15,16,bt1), (16,17,bt1), (17,18,bt1),
             (19,20,bt1), (20,21,bt1), (21,22,bt1)]
# Make a list of the bond types, coefficients and lengths
self.createBondSpec([2*6309.47]*4)
```

Listing 13: Specification of a coarse-grained asphaltene model: the bonds

```
# Then set up the angles
self.NrAngleTypes = 2
at1, at2 = self.getAngleNames()
# Specify which angles are which types
self.Angles =[ ( 0,  1,  2, at1), ( 1,  2,  3, at1),
               ( 3,  6, 10, at2), ( 4,  5,  6, at2), ( 4,  8, 12, at2),
               ( 5,  6,  7, at2), ( 5,  9, 13, at2), ( 6, 10, 14, at2),
               ( 8,  9, 10, at2), (11, 12, 13, at2), (12, 13, 14, at2),
               (10,  6,  3, at2), ( 8, 12, 15, at2), (13, 14, 19, at2),
               (15, 16, 17, at1), (16, 17, 18, at1),
               (19, 20, 21, at1), (20, 21, 22, at1) ]
# Then create the data structure. Pass in a list of tuples containing
# the angle constant and equilibrium angle for each type of angles.
self.createAngleSpec([(AlkaneAngleConstant,AlkaneAngleZero),(3e5,math.pi)])
self.Citation = "DOI: 10.4043/26155-MS"
```

Listing 14: Specification of a coarse-grained asphaltene model: the angles

field parameters, together with a listing that contains a raaSAFT implementation of the result. This listing can be placed in a file in the `mysaft` folder of raaSAFT and then imported and used in a jobscript without any modifications. Together, Bottled SAFT and raaSAFT offer an unprecedented ease-of-use for obtaining force field parameters and setting up simulations.

### *3.7. The backend code*

Some aspects of the backend code may be of general interest. In particular, raaSAFT contains an interesting citation feature, something which has become popular in scientific codes (e.g. [6], [29], [19]). This feature basically asks[1] the user to cite the relevant papers

[1] It may be interesting to note at this point that *demanding* a citation, particularly in the code's license, is not compatible with the accepted definition of open-source [30].

in a publication that uses the simulation tool. The feature may also provide a file with references e.g. in BibTeX format. For some codes, the list of citations is short and rarely changes, so it is fine to hard-code the citations into the code. But for other codes the list changes frequently, as in raaSAFT where this list changes whenever a new compound is described in the literature for the first time, so a more flexible approach was needed.

As seen in Section 3.5, the compounds in raaSAFT have a citation attribute which holds the DOI (Digital Object Identifier) of the paper describing that compound, if a paper has been published about it. The citation feature in raaSAFT takes these DOIs and downloads from the web the appropriate BibTeX block for each DOI. There are three functions which accomplish this in the following manner.

```
def listDOIs(components):
    compdoi = [("General method","10.1146/annurev-chembioeng-061312-103314"),
               ("Combination rules","10.1080/00268976.2012.662303")]
    for comp in components
        trydoi = re.search("10.[0-9]{3,}\S+", comp.Citation)
        if trydoi != None and not any(comp.Name == a for (a,b) in compdoi):
            compdoi.append((comp.Name,trydoi.group(0)))
    return compdoi

def doi2bib(doi):
    url = "http://dx.doi.org/" + doi
    r = requests.get(url,headers={"accept":"application/x-bibtex"})
    r.encoding = "UTF-8"
    return r.text

def dumpBibTeX(compdoi):
    with open('raaSAFT.bib', 'w') as bibfile:
      for cite in compdoi:
        bibfile.write(cite[0]+"\n"+doi2bib(cite[1])+"\n")
```

Listing 15: Functions for converting the valid DOIs in the system to a corresponding BibTeX file.

The first function, `listDOIs()`, builds a list of the name and DOI of all components in the system, checks that the DOI is valid, and ensures that components are not repeated. Two general references are placed first. The second function, `doi2bib()`, accepts a string containing a valid DOI, constructs the full URL for a DOI lookup from this, and sends a `GET` HTML request to this URL with the header set to ask for a BibTeX response. The correct character encoding is then set on the response, and the text in this response (which consists of the BibTeX block) is returned. The third function, `dumpBibTeX()`, takes the output from the first function, filters the DOIs through the second function, and then writes the result to a file. The compound name is put before each BibTeX block, it then becomes a comment in the BibTeX file.

To hook this functionality into HOOMD-blue's `cite.save()` command, we employ a decorator, a concept from functional programming languages. Since functions are first-class members of the Python language, another function may accept a function and return a function. This can be used to modify library functions, a procedure informally known as

monkey patching. In raaSAFT, we also check whether the `cite.save()` function exists, and if it does not (for earlier HOOMD-blue versions) we provide it in raaSAFT. This means the user jobscript does not have to change when running with different versions of HOOMD-blue. The `context.initialize()` command recently introduced in HOOMD-blue is monkey-patched in a similar fashion, to avoid having users change their jobscripts based on the HOOMD-blue version. This is very convenient if one is running on several machines, e.g. setting up jobs and doing the initial run on a workstation, and then doing the long production runs on a cluster which may have an older HOOMD-blue version installed.

## 4. Performance and applications

In this section we present a weak scaling study on up to 8 GPUs using HOOMD-blue as the backend. We then show the results from a more complicated simulation of a polymer in an explicit solvent, and finally we study the strong scaling on up to 24 GPUs using this system.

### *4.1. Weak scaling*

The test case we consider here is a simple simulation of liquid toluene, represented by two beads per molecule, so it has a bond. The weak scaling was tested on a machine having two eight-core Intel Xeon E5-2620 v2 CPUs, 24 GB of RAM, and eight Nvidia Tesla K40 GPUs. For technical reasons, four GPUs are attached to each CPU, and the CPUs communicate with each other over a dual-rail QDR InfiniBand network (Mellanox ConnectX-3 adapters). The case for the single-GPU run consisted of 62 500 toluene molecules, and the larger simulations were run with the same system replicated in the z-direction, up to 500 000 molecules on 8 GPUs. With 125 000 beads per GPU, the results by [7] indicate that the parallel efficiency should be above 80%. The case set up for two GPUs was also run in GROMACS using the exact same force field parameters and settings. This was run on just the two CPUs of the machine, since GROMACS does not support using the Mie potential on GPUs at the time of writing.

To compare the scaling, the performance metric used was "million atom-nanoseconds/hour", i.e. if the system contains one million atoms, how many nanoseconds can the code simulate per hour of walltime. We plot this metric against the number of GPUs in Figure 3. It is seen that going from 1 to 2 GPUs incurs a performance hit, but after this scaling is excellent.

### *4.2. Application: the dynamics of polystyrene in solution*

Molecular simulations of polymers present formidable challenges, since both the time and length scales of interest exceed what is commonly attained with fully detailed atomistic models. Furthermore, the fine detail is often irrelevant to the macroscopic physics. Here, again, there is value in considering coarse grained simulations. A model of polystyrene has been proposed in [31] based on an alkane-like chain "decorated" with dangling aromatic-like beads. The model is tuned to reproduce both the correct thermophysical properties of polystyrene oligomers and the liquid-liquid phase behaviour in mixtures of solvents.

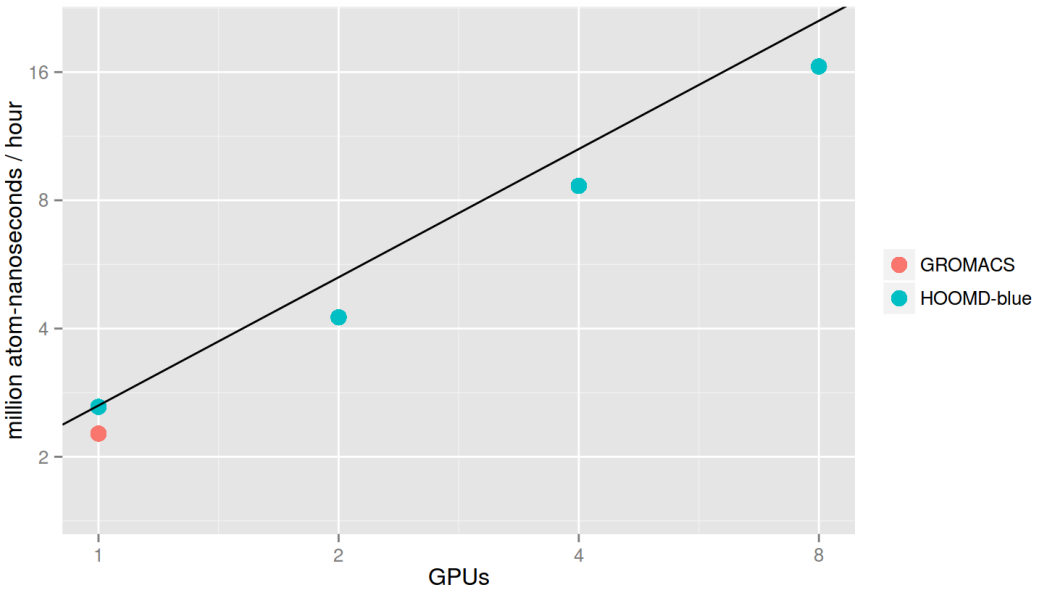



Figure 3: Weak scaling of liquid toluene system, toluene modelled as two beads at $T = 293$ K and $P = 1$ bar. Simulations run on one to eight GPUs with HOOMD-blue, and comparison with GROMACS on two six-core CPUs.

To showcase how raaSAFT can be used for such large simulations with complicated molecules, we consider as an application the phase equilibria of polystyrene in an explicit solvent, namely heptane. The simulation encompasses 294 polystyrene molecules, modelled using 46 repeating units formed from two distinct beads, one representing the aliphatic and one representing the aromatic moieties. The total molecular weight of one such molecule is 4791 g/mol. These molecules are surrounded by 42 700 molecules of a two-bead model of heptane. The system contains a total of 112 400 beads, and the molar fraction of solvent is 0.76.

We simulate this system at a pressure of $P = 30$ bar and at three different temperatures, $T_1 = 310$ K, $T_2 = 420$ K and $T_3 = 530$ K. Experiments reported in [32] show that this system has a single phase at the middle temperature $T_2$, i.e. the polystyrene is soluble in the heptane at this temperature, while the polystyrene is not soluble at the higher and lower temperatures $T_1$ and $T_3$.

The simulation box was set up as described in Section 3.3, and run in the NPT ensemble to generate three initial systems at the desired temperatures and pressure. The time step was increased from 0.001 ps to 0.01 ps after this. To obtain an equilibrated system, the simulations were then run in the NVT ensemble, and the polystyrene-polystyrene energies were monitored as the simulations progressed. These energies are plotted in Figure 4. We observe that the single-phase system at $T_2$ equilibrates very quickly, while the phase-separated systems at $T_1$ and $T_3$ take much longer, in the case of $T_1$ several hundred nanoseconds. The polystyrene implementation and the jobscripts to replicate these simulations are shown in the Appendix in Listing 17–Listing 19 and will be included in the `replication` folder of the raaSAFT repository. Note that the entire polystyrene implementation spans less than 50 lines of code, as do

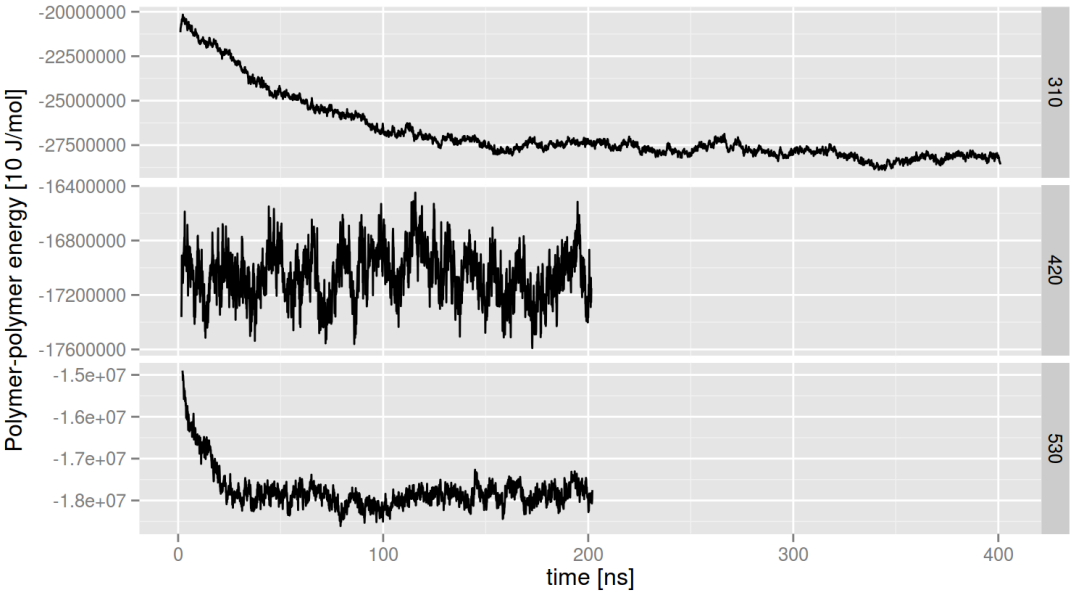



Figure 4: Polymer-polymer energies versus time, sampled every 10 ps as the simulations progressed in the NVT ensemble. From top to bottom, the temperatures are 310, 420, 530 K. It was not clear that the system at 310 K was equilibrated after 200 ns, so it was run for twice as long as the other temperatures.

the jobscripts.

The current simulation is equivalent to an atomistically-detailed simulation with 1.2 million atoms running for 400 nanoseconds. For each coarse-grained bead there would be $\sim$ 10 atomistic centres, and the computing time scales with the square of this number. Since the atomistically detailed model would also have to incorporate electrostatic interactions, and since coarse-grained simulations have a tendency to explore the configuration space (and thus reach equilibrium) faster than atomistic simulations [33], a conservative estimate is that it would require computational resources three orders of magnitude larger than those employed here. A three-orders-of-magnitude speedup is the same as that reported in [34], where the atomistic model is of the united-atom type, i.e. combining a single carbon atom with the hydrogen atoms attached to it into one bead. This suggests that the speedup over a fully atomistically detailed simulation, accounting also for the individual hydrogen atoms, is probably closer to four orders of magnitude.

The 400 nanosecond simulation here took 135 hours of walltime when running on four Nvidia Tesla K20 GPUs. Since the strong scaling is limited by the domain decomposition becoming inefficient when domain sizes become comparable to the force field cutoff, an equivalent atomistic simulation would be able to effectively utilise at most $\sim$ 40 GPUs. With the conservative estimate of the coarse-grained speedup being three orders of magnitude, the atomistic simulation would thus take *more than a year* of walltime.

A snapshot from the simulation is shown in Figure 5, indicating the complexity of this system. The heptane is invisible, and the polystyrene is shown as dark purple beads for

the aliphatic backbone and light purple beads for the aromatic-like moieties. The top half shows the system at $T_1$, where the polystyrene is clearly not soluble, while the bottom half shows the system at $T_2$, where it is soluble. The snapshot for $T_3$ is similar to that for $T_1$, and thus the model reproduces the phase behaviour seen in experiments.

To confirm these visual impressions, we have computed histograms of the solvent density in sub-cells of the simulation box, shown in Figure 6. The simulation box is divided into $6^3$ cells, and the molar fraction of solvent is calculated in each box. This calculation is averaged over 300 configurations each taken 10 000 time steps (100 picoseconds) apart, with the equilibrated system. The plot shows that for the solvated system at $T_2$, the distribution of molar fractions is unimodal and centered around the total system molar fraction. For the temperatures $T_1$ and $T_3$, however, the distributions are clearly bimodal, showing that some cells have a low solvent fraction and others have a high solvent fraction. This is indicative of phase-separated systems at these two temperatures.

### *4.3. Strong scaling*

To test further the strong scaling, now on a complex system, we have measured the performance of the polystyrene simulation discussed in the previous section, at $P = 30$ bar and $T = 310$ K for the equilibrated system, varying the number of GPUs from 1 to 24. These simulations are run on a larger cluster, where each node has two 4-core Intel Xeon E5-2609 CPUs, 64 GB of RAM, and two Nvidia Tesla K20Xm GPUs. The nodes are connected by a single-rail FDR InfiniBand network (Mellanox ConnectX-3 adapters).

The scaling is shown in Figure 7, which indicates a good strong scaling. At 24 GPUs, the number of beads per GPU is becoming too small to have good performance, so the scaling levels off. These results are very similar to those reported in [7, Fig. 9]. Fig. 8 in that publication demonstrates that for even bigger systems with more beads, one can make efficient use of thousands of GPUs.

## 5. Conclusions

In this paper we have shown raaSAFT to be a easy-to-use and flexible framework for implementing coarse-grained molecular dynamics models with the SAFT-$\gamma$ Mie theory. Models for many common substances are included with the code, and parameters for a very large number of substances can be downloaded from our online database Bottled SAFT. More complicated models, which may be built from simpler models with a group-contribution approach, are easily implemented thanks to the flexible underlying datastructures. The resulting models may be run with existing molecular dynamics codes HOOMD-blue or GROMACS. Utilising the excellent scalability of HOOMD-blue, we have demonstrated good weak and strong scaling on a multi-GPU machine and on a larger cluster with GPUs.

## Acknowledgements

The authors are grateful to Dr. Tom Headen for running the GROMACS simulation used as a reference in the weak scaling test, and to Dr. Carmelo Herdes for input

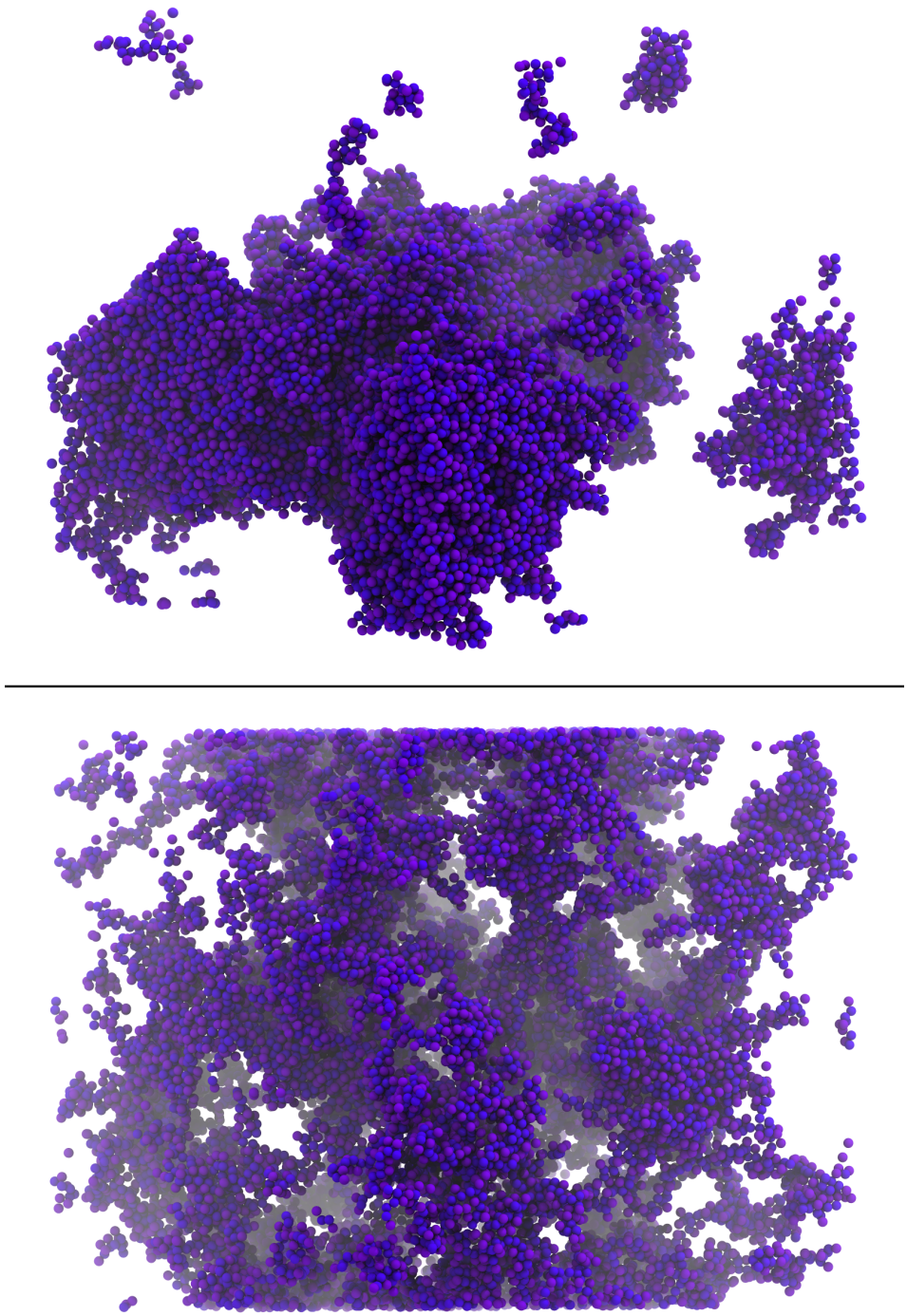


Figure 5: Snapshots of equilibrated configurations from a simulation of polystyrene in heptane at 310 K (top) and 420 K (bottom). Heptane molecules are not shown for clarity. It is seen clearly that the polystyrene is dissolved at 420 K while it is clustered at 310 K. At the top of the figure, individual polymers can be seen. The individual beads seen close to the edges are due to the periodic boundary conditions.

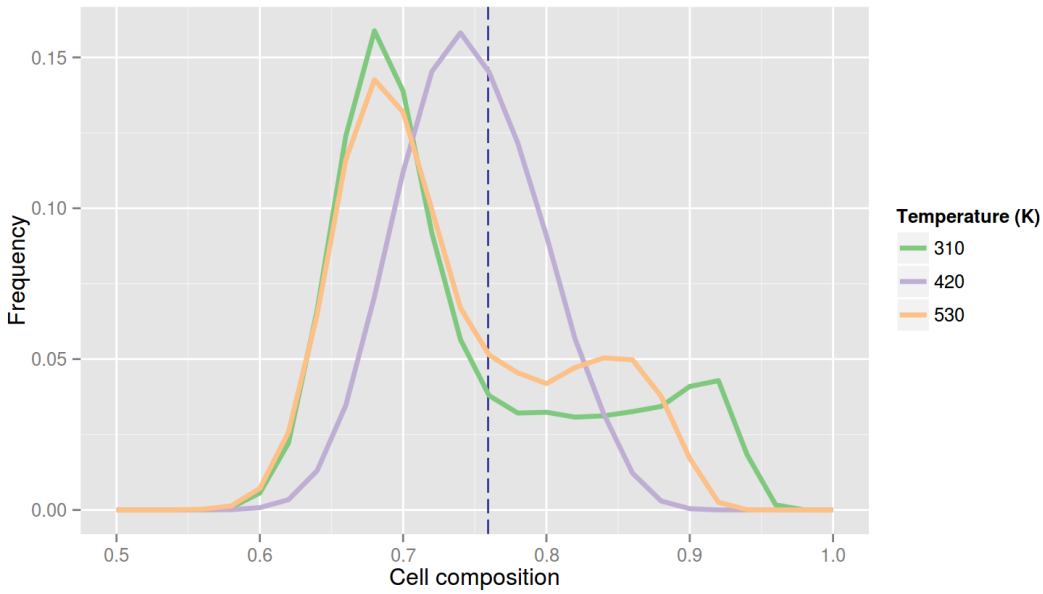



Figure 6: Quantitative evaluation of phase splits in finite systems, following [35]. Dividing the simulation box into $6^3$ cells, we compute the compositions $N_{\text{solvent}}/N_{\text{total}}$ for all cells and plot a histogram of these. The dotted line is the composition for the whole simulation box. From the unimodal versus bimodal distributions, it is apparent that the polystyrene is solvated at 410 K while it is insoluble at 310 K and 530 K, in agreement with experiments.

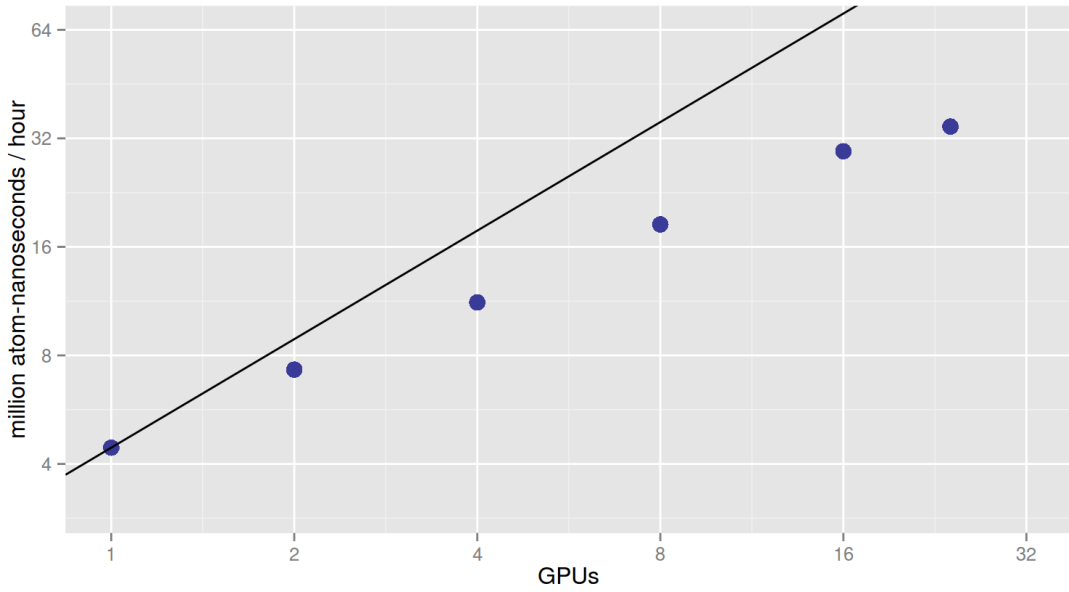



Figure 7: Strong scaling of the polymer system. Simulations run on 1 to 24 GPUs with HOOMD-blue. The scaling is very good initially, and then starts to level off at 24 GPUs, when the number of beads per GPU is becoming small.

on the GROMACS backend and the development of the model asphaltene. Prof. Bernhard Müller and Dr. Svend Tollak Munkejord are thanked for their comments on the manuscript. We also thank Dr. Joshua A. Anderson for helpful discussions around HOOMD-blue.

ÅE acknowledges the financial support from the project *Fundamental understanding of electrocoalescence in heavy crude oils* coordinated by SINTEF Energy Research and funded by the Petromaks programme of the Research Council of Norway (206976), Petrobras, Statoil and Wärtsilä Oil & Gas Systems.

GJS and EAM acknowledge the financial support for the Molecular Systems Engineering group from the Engineering and Physical Sciences Research Council (grant numbers EP/E016340, EP/J014958, EP/L020564). GJS also acknowledges the Skolkovo Foundation and BP for financial support through the UNIHEAT project.

None of the funding sources listed above were involved in the design, execution, report writing or decision to submit this work.

We are also grateful for computational resources at the Imperial College High Performance Computing Service, and at the Abel cluster operated by the University of Oslo and the Norwegian metacenter for High Performance Computing (NOTUR), as well as assistance from the staff at both locations.

## References

[1] E. Brini, E. A. Algaer, P. Ganguly, C. Li, F. Rodriguez-Ropero, N. F. A. van der Vegt, Systematic coarse-graining methods for soft matter simulations - a review, Soft Matter 9 (2013) 2108–2119.

[2] S. J. Marrink, H. J. Risselada, S. Yefimov, D. P. Tieleman, A. H. De Vries, The MARTINI force field: coarse grained model for biomolecular simulations, The Journal of Physical Chemistry B 111 (2007) 7812–7824.

[3] E. A. Müller, G. Jackson, Force-field parameters from the SAFT-$\gamma$ equation of state for use in coarse-grained molecular simulations, Annual review of chemical and biomolecular engineering 5 (2014) 405–427.

[4] C. G. Aimoli, E. J. Maginn, C. R. Abreu, Force field comparison and thermodynamic property calculation of supercritical CO2 and CH4 using molecular dynamics simulations, Fluid Phase Equilibria 368 (2014) 80 – 90.

[5] N. Ramrattan, C. Avendaño, E. Müller, A. Galindo, A corresponding-states framework for the description of the Mie family of intermolecular potentials, Molecular Physics (2015) 1–16.

[6] J. A. Anderson, C. D. Lorenz, A. Travesset, General purpose molecular dynamics simulations fully implemented on graphics processing units, Journal of Computational Physics 227 (2008) 5342 – 5359.

[7] J. Glaser, T. D. Nguyen, J. A. Anderson, P. Lui, F. Spiga, J. A. Millan, D. C. Morse, S. C. Glotzer, Strong scaling of general-purpose molecular dynamics simulations on GPUs, Computer Physics Communications 192 (2015) 97 – 107.

[8] H. Berendsen, D. van der Spoel, R. van Drunen, GROMACS: A message-passing parallel molecular dynamics implementation, Computer Physics Communications 91 (1995) 43 – 56.

[9] D. N. LeBard, Advancing simulations of biological materials: applications of coarse-grained models on graphics processing unit hardware, Molecular Simulation 40 (2014) 802–820.

[10] C. Avendaño, T. Lafitte, A. Galindo, C. S. Adjiman, G. Jackson, E. A. Müller, SAFT-$\gamma$ force field for the simulation of molecular fluids. 1. A single-site coarse grained model of carbon dioxide, The Journal of Physical Chemistry B 115 (2011) 11154–11169.

[11] C. Avendaño, T. Lafitte, C. S. Adjiman, A. Galindo, E. A. Müller, G. Jackson, SAFT-$\gamma$ force field for the simulation of molecular fluids: 2. Coarse-grained models of greenhouse gases, refrigerants, and long alkanes, The Journal of Physical Chemistry B 117 (2013) 2717–2733.

[12] T. Lafitte, C. Avendaño, V. Papaioannou, A. Galindo, C. S. Adjiman, G. Jackson, E. A. Müller, SAFT-$\gamma$ force field for the simulation of molecular fluids: 3. Coarse-grained models of benzene and hetero-group models of n-decylbenzene, Molecular Physics 110 (2012) 1189–1203.

[13] O. Lobanova, C. Avendaño, T. Lafitte, E. A. Müller, G. Jackson, SAFT-$\gamma$ force field for the simulation of molecular fluids: 4. A single-site coarse-grained model of water applicable over a wide temperature range, Molecular Physics (2015) 1–22.
[14] O. Lobanova, A. Mejía, G. Jackson, E. A. Müller, SAFT-$\gamma$ force field for the simulation of molecular fluids 6: Binary and ternary mixtures comprising water, carbon dioxide, and n-alkanes, The Journal of Chemical Thermodynamics 93 (2016) 320 – 336.
[15] A. Mejía, C. Herdes, E. A. Müller, Force fields for coarse-grained molecular simulations from a corresponding states correlation, Industrial & Engineering Chemistry Research 53 (2014) 4131–4141.
[16] Å. Ervik, A. Mejía, E. A. Müller, Bottled SAFT: a web app providing SAFT-$\gamma$ Mie force field parameters for thousands of molecular fluids, In preparation (2016).
[17] V. Papaioannou, T. Lafitte, C. Avendaño, C. S. Adjiman, G. Jackson, E. A. Müller, A. Galindo, Group contribution methodology based on the statistical associating fluid theory for heteronuclear molecules formed from Mie segments, The Journal of Chemical Physics 140 (2014).
[18] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, P. Gumbsch, Structural relaxation made simple, Phys. Rev. Lett. 97 (2006) 170201.
[19] W. Humphrey, A. Dalke, K. Schulten, VMD – Visual Molecular Dynamics, Journal of Molecular Graphics 14 (1996) 33–38.
[20] B. Hess, Determining the shear viscosity of model liquids from molecular dynamics simulations, The Journal of Chemical Physics 116 (2002).
[21] S. Zeppieri, J. Rodríguez, A. L. L. de Ramos, Interfacial tension of alkane + water systems, Journal of Chemical & Engineering Data 46 (2001) 1086–1088.
[22] T. Giorgino, Computing 1-D atomic densities in macromolecular simulations: The density profile tool for VMD, Computer Physics Communications 185 (2014) 317 – 322.
[23] A. Mejía, M. Cartes, H. Segura, E. A. Müller, Use of equations of state and coarse grained simulations to complement experiments: Describing the interfacial properties of carbon dioxide + decane and carbon dioxide + eicosane mixtures, Journal of Chemical & Engineering Data 59 (2014) 2928–2941.
[24] E. A. Müller, C. Herdes, T. S. Totton, Coarse-grained models for crude oils: A direct link between equations of state and molecular simulations, in: Offshore Technology Conference.
[25] M. Subirana, E. Y. Sheu, Asphaltenes: fundamentals and applications, Springer Science & Business Media, 2013.
[26] O. C. Mullins, H. Sabbah, J. Eyssautier, A. E. Pomerantz, L. Barré, A. B. Andrews, Y. Ruiz-Morales, F. Mostowfi, R. McFarlane, L. Goual, R. Lepkowicz, T. Cooper, J. Orbulescu, R. M. Leblanc, J. Edwards, R. N. Zare, Advances in asphaltene science and the Yen–Mullins model, Energy & Fuels 26 (2012) 3986–4003.
[27] A. M. McKenna, L. J. Donald, J. E. Fitzsimmons, P. Juyal, V. Spicer, K. G. Standing, A. G. Marshall, R. P. Rodgers, Heavy petroleum composition. 3. Asphaltene aggregation, Energy & Fuels 27 (2013) 1246–1256.
[28] P. K. Kilpatrick, Water-in-crude oil emulsion stabilization: Review and unanswered questions, Energy & Fuels 26 (2012) 4017–4026.
[29] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, Efficient management of parallelism in object-oriented numerical software libraries, in: Modern Software Tools for Scientific Computing, Springer, 1997, pp. 163–202.
[30] Debian Legal Team, Debian free software guidelines and software license FAQ, 2015. `http://people.debian.org/~bap/dfsg-faq.html`.
[31] M. G. Jiménez-Serratos, A. Haslam, G. Jackson, E. A. Müller, Group contribution modelling of polystyrene using SAFT-$\gamma$ force fields. Molecular simulations of melts and polystyrene solvent mixtures, In preparation (2016).
[32] J. Cowie, I. McEwen, Polymer-cosolvent systems: 5. Upper and lower critical solution temperatures of polystyrene in n-alkanes, Polymer 24 (1983) 1445 – 1448.
[33] C. Herdes, T. S. Totton, E. A. Müller, Coarse grained force field for the molecular simulation of natural gases and condensates, Fluid Phase Equilibria 406 (2015) 91 – 100.
[34] V. A. Harmandaris, N. P. Adhikari, N. F. A. van der Vegt, K. Kremer, Hierarchical modeling of polystyrene: From atomistic to coarse-grained simulations, Macromolecules 39 (2006) 6708–6719.
[35] F. Martínez-Veracoechea, E. Müller, Temperature-quench molecular dynamics simulations for fluid phase equilibria, Molecular Simulation 31 (2005) 33–43.

## Appendix A. Jobscripts and polystyrene implementation

In the following listings we present the collection of the partial listings 1–9 for the octane-water simulation in Listing 16, the implementation of the polystyrene model in Listing 17, and the jobscripts for running the polystyrene simulation in Listing 18 and Listing 19.

```
#!/usr/bin/python
from hoomd_script import *
from raasaft.main import *
from raasaft.alkanes import *
from raasaft.water import *
context.initialize()
C8 = HomoAlkane(C=8, count=1e4)
H2O = BioWater(count=6e4)
components = [C8,H2O]
theBox = setupSimBox(components, elong=3.0, packing=0.1)
system = setupSimulation(components, theBox)
setCutoff(components, cutoff=20)
setCrossInteraction(C8, H2O, kij=0.3205)
table = pair.table(width=1000)
setPotentialCoeffs(components, table)
setAllCrossCoeff(components, table)
setupBonds(system, components)
setupAngles(system, components)
all = group.all()
mode = integrate.mode_standard(dt=0.001)
relaxation = integrate.nve(group=all, limit=0.01)
run(1e3)
relaxation.disable()
Temp = 298.15*kBby10
nvt = integrate.nvt(group=all, T=Temp, tau=0.5)
run(4e4)
nvt.disable()
npt=integrate.npt(group=all, T=Temp, P=1*ConvFromBar, tau=0.5, tauP=0.5)
run(1e5)
periodic = external.periodic()
periodic.force_coeff.set(C8.Name, A = 400.0, i=2, w=0.2, p=1)
periodic.force_coeff.set(H2O.Name, A = -400.0, i=2, w=0.2, p=1)
run(1e6)
periodic.disable()
update.zero_momentum()
npt.disable()
xml = dump.xml(filename='con.xml', vis=True)
dump.dcd(filename='dump.dcd', period=1e3)
logQuant = ['time', 'pair_table_energy', 'pressure', 'temperature',
                    'pressure_xx', 'pressure_yy', 'pressure_zz']
analyze.log(filename='log.dat', quantities=logQuant, period=1e2,header_prefix='')
mode.set_params(dt=0.01)
nvt = integrate.nvt(group=all, T=Temp, tau=0.5)
run(2e6)
xml = dump.xml(filename='restart.xml', vis=True, velocity=True)
```

Listing 16: A collection of the partial listings 1–9 in Section 3.3. Note that only the lines up to and including line 18 are raaSAFT-specific.

```
from __future__ import (absolute_import, division,print_function)
from builtins import *
from raasaft.mie import *
from raasaft.constants import *
class AromaticPS(MieCG):
    def __init__(self,count):
        self.Name = "AromaticPS"
        self.Epsilon=410.273*kBby10
        self.Sigma=4.260
        self.N=16.829
        self.M=6
        self.Segments = 2
        self.Mass = 50
        self.BondConstant = RigidBondConstant
        self.Citation = "DOI: Unpublished"
        MieCG.__init__(self,count)
class BackbonePS(MieCG):
    def __init__(self,count):
        self.Name = "BackbonePS"
        self.Epsilon=377.14*kBby10
        self.Sigma=4.180
        self.N=16.430
        self.M=6
        self.Segments = 2
        self.Mass = 54.15
        self.BondConstant = RigidBondConstant
        self.Citation = "DOI: Unpublished"
        MieCG.__init__(self,count)
class Polystyrene(MieCGHet):
    def __init__(self,count,monomers):
        self.BBN = BackbonePS(count=1)
        self.BCH = BenzenePS(count=1)
        self.Components=[self.BBN, self.BCH]
        self.Name = "Polystyrene"
        self.ShortName = "PS"
        self.Segments = 2*monomers
        self.Def = ([self.BBN]+[self.BCH])*monomers
        MieCGHet.__init__(self,count)
        self.NrBondTypes = 2
        bt1, bt2, = self.getBondNames()
        self.Bonds = []
        for m in range(monomers-1):
            self.Bonds.extend( [ (2*m,2*m+1,bt1), (2*m,2*m+2,bt2) ] )
        # Handle the last bond separately, since there is no bt2 to next alkane
        self.Bonds.append((2*(m+1),2*(m+1)+1,bt1))
        self.createBondSpec([10000,6309.5])
        self.Citation = "DOI: unpublished"
```

Listing 17: Implementation of the polystyrene model. The first two classes specify the bead types, and the final class specifies the heteronuclear model. Note that the constructor for this model takes the number of monomers as input.

```
from hoomd_script import *
from raasaft.main import *
from raasaft.polymers import *
from raasaft.alkanes import *
import math
context.initialize()
C7 = MnMAlkane(C=7,count=42672)
PS = Polystyrene(monomers=46,count=294)
components = [C7,PS]
theBox = setupSimBox(components,elong=1.0,packing=0.7)
system = setupSimulation(components,theBox)
setCutoff(components,cutoff=20)
setCrossInteraction(PS.BBN,C7,kij=-0.01)
setCrossInteraction(PS.BCH,C7,kij=0.045)
table = initMiePotential(table=True)
setPotentialCoeffs(components,table)
setAllCrossCoeff(components,table)
setupBonds(system,components)
Temp = 310*kBby10
Pres = 30.0*ConvFromBar
all = group.all()
mode = integrate.mode_standard(dt=0.001)
xml = dump.xml(filename='ps-init-system.xml', vis=True)
dump.dcd(filename='ps-init-dump.dcd', period=1e3)
logQuant = ['time', 'pair_table_energy', 'pressure', 'temperature',
  'pressure_xx', 'pressure_yy', 'pressure_zz', 'lx', 'ly', 'lz']
analyze.log(filename='log.dat', quantities=logQuant, period=1e2, header_prefix='')
relaxation = integrate.nve(group=all, limit=0.01)
run(1e3)
relaxation.disable()
nvt = integrate.nvt(group=all, T=Temp, tau=0.5)
run(5e4)
nvt.disable()
mode.set_params(dt=0.005)
npt=integrate.npt(group=all, T=Temp, P=Pres, tau=0.5, tauP=0.5)
npt.set_params(T=310*kBby10)
run(5e4)
xml = dump.xml(filename='restart-after-npt-t310.xml', vis=True, velocity=True)
npt.set_params(T=420*kBby10)
run(5e4)
xml = dump.xml(filename='restart-after-npt-t420.xml', vis=True, velocity=True)
npt.set_params(T=530*kBby10)
run(5e4)
xml = dump.xml(filename='restart-after-npt-t530.xml', vis=True, velocity=True)
```

Listing 18: Generating the initial systems with correct temperatures and pressure for the polystyrene simulations.

```
1  from hoomd_script import *
2  from raasaft.main import *
3  from raasaft.polymers import *
4  from raasaft.alkanes import *
5  import math
6  context.initialize()
7  C7 = MnMAlkane(C=7,count=42672)
8  PS = Polystyrene(monomers=46,count=294)
9  components = [C7,PS]
10 system = init.read_xml(filename="restart-after-npt-t310.xml")
11 setCutoff(components,cutoff=20)
12 setCrossInteraction(PS.BBN,C7,kij=-0.01)
13 setCrossInteraction(PS.BCH,C7,kij=0.045)
14 table = initMiePotential(table=True)
15 setPotentialCoeffs(components,table)
16 setAllCrossCoeff(components,table)
17 # dummy tables for logging partial energies
18 ## for d1, keep just polystyrene and its internal cross-interactions
19 d1 = pair.table(width=1000,name="pstpst")
20 setPotentialCoeffs(components,d1,keep=[PS])
21 setAllCrossCoeff(components,d1,keep=[PS])
22 d1.disable(log=True)
23 ## for d2, keep just heptane and no cross interactions
24 d2 = pair.table(width=1000,name="hephep")
25 setPotentialCoeffs(components,d2,keep=[C7])
26 setAllCrossCoeff(components,d2,keep=[])
27 d2.disable(log=True)
28 setupBonds(system,components)
29 Temp = 310*kBby10
30 all = group.all()
31 integrate.mode_standard(dt=0.01)
32 xml = dump.xml(filename='con-t310.xml', vis=True)
33 dump.dcd(filename='dump-t310.dcd', period=1e4)
34 logQuant = ['potential_energy', 'pair_table_energy_pstpst',
35   'pair_table_energy_hephep', 'temperature', 'pressure',
36   'pressure_xx', 'pressure_yy', 'pressure_zz']
37 analyze.log(filename='log.dat', quantities=logQuant, period=1e2, header_prefix='')
38 nvt = integrate.nvt(group=all, T=Temp, tau=0.5)
39 run(40e6)
40 dump.xml(filename="final-t310.xml", vis=True, velocity=True)
```

Listing 19: Implementation of a production run that takes the initial system from Listing 18 and continues for 40 million time steps (line 39), providing for an analysis of equilibrium by studying the polystyrene–polystyrene energies. Note the syntax for logging partial energies on lines 17-27.

# Paper 7

# Bottled SAFT: a web app providing SAFT-$\gamma$ Mie force field parameters for thousands of molecular fluids.

Åsmund Ervik,[†] Andres Mejía,[‡] and Erich A. Müller[*,¶]

[†]*Department of Energy and Process Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway*

[‡]*Departamento de Ingeniería Química, Universidad de Concepción, Concepcíon, Chile*

[¶]*Department of Chemical Engineering, Imperial College London, London, UK*

E-mail: e.muller@imperial.ac.uk

**Abstract**

Coarse-grained molecular simulation is a popular tool for modelling simple and complex fluids alike. The defining aspects of a coarse grained model are the force field parameters, which must be determined for each particular fluid. Since the number of molecular fluids important in nature and in engineering processes is immense, constructing force field parameter tables manually does not scale well. An important step towards solving this challenge was taken recently by Mejia et al., who proposed a correlation that provides force field parameters for a fluid given its critical properties. The force field in question is the successful SAFT-$\gamma$ Mie approach. Building on this, we have applied the correlation to more than 6000 fluids, and constructed a web application which makes this data set easily searchable by CAS number, name or chemical formula. We call this "Bottled SAFT". Once a result has been found in Bottled SAFT, code snippets are provided for simulating the desired substance using our recently released

"raaSAFT" framework, which leverages established molecular dynamics codes to run the simulations. For a chemical not found in the database, the correlation can be run with user-provided critical properties. The code underlying the web application is written in Python using the Flask microframework; this allows us to write a modern high-performance web app while also making use of the scientific libraries available in Python. Physical intuition and experience with molecular simulation will always be required to obtain sensible simulation results. Bottled SAFT merely aims at taking the complexity out of obtaining force field parameters for a wide range of molecular fluids. The web application is freely available at `http://www.bottledsaft.org`, and the underlying source code is available on Bitbucket under a permissive license.

# Introduction

Coarse-grained molecular simulation has become a popular simulation tool. By coarse-grained we mean that the particles in a simulation do not represent individual atoms, but perhaps two to four atoms bunched together into a "bead". In comparison with atomistic simulation this trades resolution for speed; by resolving less details of the system we can simulate larger and more complicated systems at the same cost. But lower resolution is not the only difference: say there are 10 different chemical elements used in atomistically-detailed models for some class of molecules. Then the number of coarse-grained bead types that can be constructed from two, three or four of these elements is 375, and this number grows combinatorically with the number of elements. Thus coarse-grained models face a greater challenge when it comes to constructing the required force field parameters for a wide range of molecules.

A step towards solving this problem is offered by top-down coarse-graining methods, of which the SAFT-$\gamma$ Mie approach is an example; see Ref. 1 for a thorough review. An overview of the approach is given in the supporting information. In principle, this approach obtains the force field parameters by fitting an equation of state to experimental thermophysical data, and having a one-to-one link between the equation of state and the molecular force

field. This approach is powerful and does not require any trial-and-error, but still it requires some amount of computational work. Recognising that the procedure could be simplified further, Mejía et al.[2] developed the "M&M" correlation which takes as input just the critical properties and the acentric factor of the compound, as well as the desired number of beads in the molecule, and the liquid density at a reduced temperature $T_r = 0.7$; if unavailable this can be estimated from a Rackett-type equation[3]. Thus the force field parameters for a given liquid can be computed instantaneously. Furthermore, large tables with the required critical properties are found in the literature. For the current application, we have used as a basis the book by Yaws[4] which gives data for more than 7800 compounds. Applying the correlation to this data set, and filtering out models which are outside the correlation range of application, models for more than 6000 compounds were available. A description of how the data set was filtered is found in the supporting information.

The recent development of web microframeworks in Python (e.g. Flask) has enabled scientists to write web applications in a language with many scientific libraries. For building Bottled SAFT, we have used Flask and its extensions together with the libraries Pandas and RDkit. By using the Bootstrap CSS extension of Flask, we achieve a modern look which works equally well on personal computers and mobile devices. We also integrate into the application the ChemDoodle Web Components JavaScript library, and demonstrate how to get a molecule drawn by the user into an RDkit representation in the Python code for further processing. The technology stack is discussed in greater detail in the next section.

In the web application Bottled SAFT, each model is identified by the name, chemical formula and the CAS number (if it has one). The application lets the user search by either of these, including searching for parts of a name. Once the user has arrived at the desired model, Bottled SAFT constructs an input file for `raaSAFT`[5], our open-source Python framework for easily setting up molecular dynamics simulations with the SAFT-$\gamma$ Mie potential. `raaSAFT` employs widely-used molecular dynamics codes to run the actual simulations; the default is HOOMD-blue[6], but GROMACS[7] can also be used. In addition to searching the database,

users may also run the M&M correlation with custom input, e.g. if they have thermophysical properties for a compound of interest which is not in the database. Finally, we provide as a proof-of-concept a page with a molecule drawer integrated with the Python backend. The name Bottled SAFT stems from the fact that "saft" in German and Scandinavian languages means "fruit juice", while the acronym SAFT stands for "Statistical Associating Fluid Theory".

# Implementation

In the following subsections we describe the implementation of the Bottled SAFT web application, detailing the roles played by the individual components. The address of the web application is `http://www.bottledsaft.org`. The code running this web application can be obtained from the repository at `https://bitbucket.org/asmunder/bottledsaft`. Executing the `saftflask.py` file will run the web application on the local computer, assuming the dependencies are installed. For a description of how the user interacts with the application, see the supporting information.

## Components of the web application

To build Bottled SAFT we have used Python 3 and we employ the Python packages Flask[8], Pandas[9], WTForms[10] and RDkit[11]. We also make use of Bootstrap[12] and ChemDoodle Web Components[13]. While the Flask framework has a built-in web server which is useful for testing, this is not usable for production. We use the combination of nginx and uWSGI to serve the application and to handle caching of content. Using Locust to test the performance, we find that the application can sustain 70 000 database lookups per hour; see the supporting information for details. We now describe these pieces and how they fit together in the application.

Flask is a web microframework written in Python. It forms the backend of our application, and is responsible for taking user input from forms and computing the appropriate output,

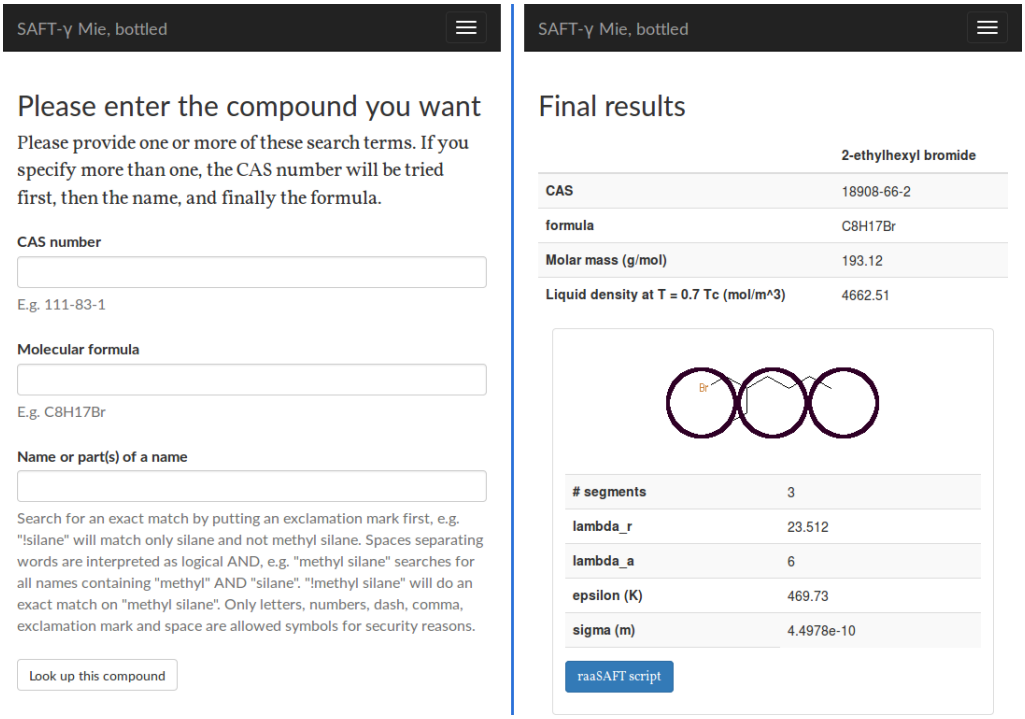



Figure 1: Left: the `/search` page implemented by the code blocks in listings 1–3. Right: the top table displays the properties of the compound. The bottom cards display the resulting force field parameters, with an image showing the number of beads superimposed on the molecule. At the bottom is a button for getting the script implementing it in raaSAFT. More cards for higher bead numbers are below this (not shown). (For higher resolution and more details please visit the actual webpage.)

for rendering the web pages requested by the user, and for routing a request to the correct page. Flask provides a RESTful interface[14]. This means that the user's web browser (the client) is responsible for storing the state of the application, in the form of cookies. This makes it easy to handle many simultaneous requests. In Flask, a request is routed using a function decorator. Thus we write our data processing as usual Python functions, with one exception: we cannot use input arguments to the function like we normally would. Instead, the input data comes either from a global variable, a form variable, or from a session cookie (i.e. data stored by the client). See the example in listing 1.

To handle form input, we use the Flask extension for the WTForms Python library. This

**Listing 1** A function that handles requests for `/search`. Both requesting the webpage (GET) and submitting the form (POST) is handled by this function.

```
@app.route('/search', methods=('GET','POST'))
def search():
  # Get the form from code in listing 2.
  form = CompoundForm(request.form)
  if request.method == 'GET':
    # Render HTML, including form, and return it
    return render_template('search.html',form=form)
  if request.method == 'POST' and form.validate():
    # If user provided CAS number, search for it
    if form.cas.data != "":
        resFrame = df[df['CAS'] == form.cas.data]
    # Search by name/formula omitted for brevity
    # Store result as JSON in a session cookie
    session['result_compound'] = resFrame.to_json()
    # Pass user on to the next page
    return redirect('/refine')
```

enables e.g. sanitising of user input with a simple syntax. See the supporting information for further details. An example is shown in listing 2.

**Listing 2** Defining the form on the `/search` page for searching the database.

```
class CompoundForm(Form):
  cas = TextField('CAS number',\
    description='E.g. 115-07-1',\
    validators=[Regexp("\d{2,7}-\d\d-\d",
                message="Not a valid CAS")]
  )
  # More fields go here, omitted for brevity
  Lookup = SubmitField('Look up this compound')
```

To render web pages, Flask uses the Jinja2 templating engine, which lets us write templates that are parsed into HTML before they are served. This makes it very simple to have HTML with dynamic parts. The Jinja2 templating language is object-oriented, meaning templates can inherit from other templates, and may override functions they have inherited. In the example in listing 3, the template inherits from the parent "saftstyle.html" template, which defines layout, fonts etc. The page heading and text is standard HTML, and the template then inserts the form provided as input to `render_template()` in the final line of listing 1.

In order to make the web application work and appear consistent across both mobile devices

and PCs, the popular Bootstrap[12] frontend framework is employed. For more descriptions of this and other implementation details, see the supporting information. The final HTML template which renders the search page is shown in listing 3.

**Listing 3** The template which is rendered to produce the HTML file for `/search`.

```
{% extends "saftstyle.html" %}
{% block content %}
<div class="container">
<div class="row">
<div class="col-md-10">
  <h2>Please enter the compound you want</h2>
  <p class="lead">
  Please provide one or more of these search
  terms. If you specify more than one, the
  CAS number will be tried first, then the
  name, and finally the formula.
  </p>
  {{wtf.quick_form(form,form_type='horizontal',
     button_map={'submit_button': 'primary'})}}
</div>
</div>
</div>
{% endblock %}
```

A final element in Bottled SAFT is the ChemDoodle molecule drawer which is integrated through JavaScript code that sends the drawn molecule back to the Python code. For details see the supporting information.

## Results and Discussion

To demonstrate how Bottled SAFT enables the study of a vast number of complicated fluid systems, we consider the ternary mixture of sulfolane, octane and benzene. This system has important applications in the chemical processing industry, where sulfolane is used for liquid-liquid extraction of aromatics during petroleum refinement. Sulfolane is also used to purify natural gas in the sulfinol process. The sulfolane + octane + benzene system has a liquid-liquid phase split where an octane-rich phase and a sulfolane-rich phase are formed, with the benzene soluble in both phases.

To be able to predict the two liquid phase compositions for the ternary system and subsequently compare the results with experiments, we consider the system at a temperature of 403.15 K, atmospheric pressure, and with an overall composition of 40 % sulfolane, 40 % octane and 20 % benzene (all in mole percent). Experimental results for the two liquid phase compositions at this system state are available due to Lee and Kim [15]. Models for this system were obtained from Bottled SAFT; for details see the supporting information.

The system was simulated using a 1:1:3 elongated simulation domain, such that the interfaces will form along the two shorter ($x$ and $y$) directions. The system contains 8000 molecules of benzene, 16 000 molecules of octane and 16 000 molecules of sulfolane, corresponding to a total of 752 000 atoms. The simulation was performed first briefly (0.1 ns) in the $NpT$ ensemble to obtain the desired temperature and pressure, and then for 40 ns in the $NVT$ ensemble to obtain equilibrium between the two liquid phases that spontaneously separate. Running on a single desktop-class GPU (an Nvidia GTX 970) the simulation took 24 hours. The molar fractions along the elongated ($z$) direction, together with a snapshot from the simulation, are shown in figure S1 in the supporting information. To compare with experiments, the bulk compositions obtained are plotted in a ternary diagram together with the experimental results from [15], figure 2, where excellent agreement is seen.

## Conclusions

Coarse-grained molecular simulation is a powerful tool, but requires a large number of force field parameters in order to cover the chemical space interesting for scientific and engineering applications. With the SAFT-$\gamma$ Mie force field, a top-down approach to coarse-graining, we can compute force field parameters directly from the critical properties of the compound in question, by use of the M&M correlation. For the web application demonstrated here, Bottled SAFT, we have applied the correlation to a large database of critical properties. This gives Bottled SAFT a database with parameters for over 6000 compounds that can be simulated.

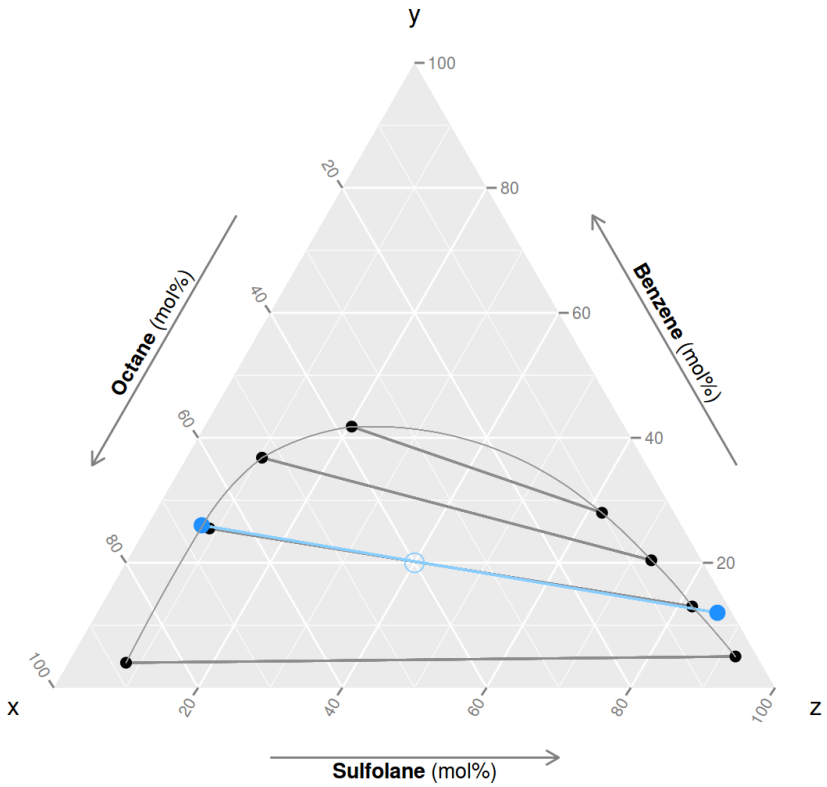



Figure 2: A ternary phase diagram, showing the experimental results by Lee and Kim [15] (black points) connected with tie-lines (dark gray), and with a thin gray curve as a guide to the eye showing the two-phase region. The blue open point in the middle is the initial composition in the simulation, and the two solid blue points are the compositions of the two liquid phases which separate. Note that the blue tie-line obscures the gray tie-line for these points.

To facilitate running molecular dynamics simulations with the resulting parameters, the user can download a script implementing the result in our raaSAFT framework. The end result is an unprecedented ease-of-use up to the point of running an initial simulation. From there on, the usual process of running molecular dynamics applies.

## Acknowledgement

The authors acknowledge the contributions of Muhammed Jansi and Karson Wong, who compiled the initial table of parameters. We also thank Prof. Bernhard Müller and Dr. Svend Tollak Munkejord for their comments on the manuscript.

ÅE acknowledges the financial support from the project *Fundamental understanding of electrocoalescence in heavy crude oils* coordinated by SINTEF Energy Research and funded by the Petromaks programme of the Research Council of Norway (206976), Petrobras, Statoil and Wärtsilä Oil & Gas Systems. EAM acknowledges financial support for the Molecular Systems Engineering group from the Engineering and Physical Sciences Research Council (grant numbers EP/E016340, EP/J014958, EP/L020564).

## Supporting Information Available

Further details are available in the Supplementary Information. These include details of the proceedure to filter the initial data set, details of the inclusion of the ChemDoodle molecule drawer, as well as details concerning the example ternary liquid-liquid equilibrium. This material is available free of charge via the Internet at `http://pubs.acs.org/`.

## References

(1) Müller, E. A.; Jackson, G. Force-Field Parameters from the SAFT-$\gamma$ Equation of State for Use in Coarse-Grained Molecular Simulations. *Annual review of chemical and*

*biomolecular engineering* **2014**, *5*, 405–427.

(2) Mejía, A.; Herdes, C.; Müller, E. A. Force fields for coarse-grained molecular simulations from a corresponding states correlation. *Industrial & Engineering Chemistry Research* **2014**, *53*, 4131–4141.

(3) Rackett, H. G. Equation of state for saturated liquids. *Journal of Chemical and Engineering Data* **1970**, *15*, 514–517.

(4) Yaws, C. L. *Thermophysical properties of chemicals and hydrocarbons*; William Andrew: Norwich, NY, 2008.

(5) Ervik, Å.; Serratos, M. G. J.; Müller, E. A. raaSAFT: a framework for easy coarse-grained molecular dynamics simulations based on the SAFT-$\gamma$ Mie force field. *In preparation* **2015**,

(6) Anderson, J. A.; Lorenz, C. D.; Travesset, A. General purpose molecular dynamics simulations fully implemented on graphics processing units. *Journal of Computational Physics* **2008**, *227*, 5342 – 5359.

(7) Berendsen, H.; van der Spoel, D.; van Drunen, R. GROMACS: A message-passing parallel molecular dynamics implementation. *Computer Physics Communications* **1995**, *91*, 43 – 56.

(8) Ronacher, A. Flask (A Python Microframework). http://flask.pocoo.org/.

(9) McKinney, W. Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference. 2010; pp 51 – 56.

(10) Johansson, T.; Crasta, J. WTForms. https://github.com/wtforms/wtforms.

(11) Landrum, G. RDKit: Open-source cheminformatics. http://www.rdkit.org.

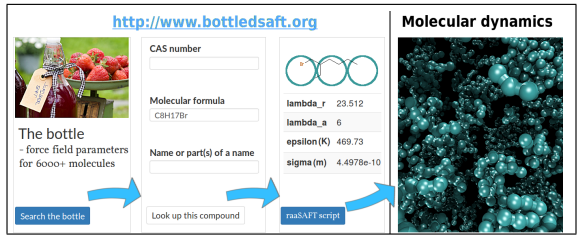
(12) Otto, M. Bootstrap - The world's most popular mobile-first and responsive front-end framework. http://getbootstrap.com/.

(13) web.chemdoodle.com, ChemDoodle Web Components | HTML5 Chemistry. https://web.chemdoodle.com/.

(14) Fielding, R. T. Architectural styles and the design of network-based software architectures. Ph.D. thesis, University of California, Irvine, 2000.

(15) Lee, S.; Kim, H. Liquid-liquid equilibria for the ternary systems sulfolane + octane + benzene, sulfolane + octane + toluene, and sulfolane + octane + p-xylene at elevated temperatures. *Journal of Chemical & Engineering Data* **1998**, *43*, 358–361.

# Graphical TOC Entry



# Supporting information for: Bottled SAFT: a web app providing SAFT-$\gamma$ Mie force field parameters for thousands of molecular fluids.

Åsmund Ervik,[†] Andres Mejía,[‡] and Erich A. Müller[*,¶]

[†]*Department of Energy and Process Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway*

[‡]*Departamento de Ingeniería Química, Universidad de Concepción, Concepcíon, Chile*

[¶]*Department of Chemical Engineering, Imperial College London, London, UK*

E-mail: e.muller@imperial.ac.uk

## The SAFT-$\gamma$ Mie approach

In a top-down method, the force field parameters are regressed from thermophysical properties of the molecule in question. With the SAFT-$\gamma$ Mie approach the top-down property comes from a one-to-one correspondence between the parameters of the molecular force field and a corresponding analytical equation of state. This means that fitting the equation of state to experimental data, e.g. to vapour and liquid densities as a function of temperature, fixes the parameters of the molecular force field. The key point, however, is that predicting thermodynamic and transport properties from molecular simulations using these force field parameters gives accurate quantitative results which are in good agreement with both the equation of state and the experimental measurements[S1].

To be specific, the force field for a given molecule is given by a chain of tangentially bonded isotropic Mie beads. The Mie potential is

$$V(r) = C(n,m)\,\epsilon\left[\left(\frac{\sigma}{r}\right)^m - \left(\frac{\sigma}{r}\right)^n\right], \tag{1}$$

$$C(n,m) = \left(\frac{m}{m-n}\right)\left(\frac{m}{n}\right)^{n/(m-n)}, \tag{2}$$

of which the commonly used Lennard-Jones potential is the special case $m = 12, n = 6$. Furthermore, it can be shown[S2] that the two exponents are related to each other, hence in this work we consider $n = 6$ as fixed. Leaving the other exponent $m$ free allows for an interplay between harder and softer potentials for different models that cannot be achieved if one sticks with the Lennard-Jones form.

## Tabulating and filtering results from the correlation

To construct the database underlying the web application, the M&M correlation was applied to each line in the table provided in Ref. S3. The results were loaded into a Python program using the Pandas library[S4,S5], with all data in a single Pandas DataFrame. Once this DataFrame was inspected, it was seen to be necessary to filter out results that were unphysical. Several criteria were applied in the filtering:

- Models where the acentric factor $\omega$ was not given in the table in Ref. S3 were excluded.
- Models where $\sigma$ was less than zero were excluded.
- Models with $m > 40$ were excluded, since these are known to give unphysically high freezing temperatures[S2].
- Models with $m < 7$ were excluded, since the potential loses the repulsive core as $m \to 6$.

## Implementation details

To handle forms, Bottled SAFT employs the WTForms library. WTForms provides many useful features, not least of which is input sanitation. For instance, the return type of a form input field is specified e.g. as a float or a string, and this is enforced by the library. We may also write functions or regular expressions for validating input. In Bottled SAFT we use a regular expression to check that the user input is a valid CAS number. If a user inputs an invalid CAS number, the web page displays an error message "Not a valid CAS" when the user tries to submit the form.

While producing HTML from templates is the purpose of the Jinja2 templating engine, we may also use it to build other text files. In Bottled SAFT we use Jinja2 to build the Python snippets that implement code for simulating the result with raaSAFT. This means we can write the boilerplate part of a Python script as a template and have Jinja2 put in the dynamic parts of the script at the users request. This is a very flexible method of dynamically building scripts or input files for the user.

To style the HTML, Bootstrap is used through the appropriate Flask plugin. Bootstrap is a responsive framework, meaning that the page layout changes according to the size of the users display. Since Bootstrap is widely employed on the web, the layout is familiar to users. With Bootstrap the layout is specified in a $12 \times 12$ grid of rows and columns, and we add `<div>` tags to our template to specify the layout of our content relative to this grid, as shown in listing 3 in the paper.

In Bottled SAFT the user input is stored to a database in two cases. The first is if the user is running the correlation for a custom compound, and has provided a reference to the literature for the values used. The second is if the user provides both a refined value for the liquid density at $0.7 \times T_c$ (as opposed to using the Rackett equation estimate) and a reference to the literature for this. The values are stored in a separate write-only database using the `HDFStore()` mechanism from the Pandas library. The user is informed clearly about this storage on the webpage.

### Using Bottled SAFT

When the user accesses `http://www.bottledsaft.org` they come to the front page. At the top of this page, three cards display short information and links to the three different components: search the database, run the correlation with custom input, or draw a molecule with the sketcher. Below this, a brief review of the SAFT-$\gamma$ Mie approach is given, with references to the literature.

When the user clicks "Search the bottle", the `/search` page is rendered from the code shown in the paper. When searching by formula or name, multiple results may be returned. The next page then displays a drop-down list where the user can select the desired compound. Once a unique compound has been selected, the user is given the option of refining the model, by entering a more accurate value for the liquid density at a reduced temperature $T_r = 0.7$. As mentioned, this value is otherwise estimated from a Rackett-type equation. Once the result has been refined, or the user has decided to use the value from the Rackett equation, the result is presented. A screenshot of this is shown in the paper. The figure of the molecule is linked in from the NIH Chemical Identifier Resolver in the form `http://cactus.nci.nih.gov/chemical/structure/<INSERT_CAS>/image`.

Once one or more models are obtained, they can be used in molecular dynamics simulations. When several models are used together in a simulation, the cross-interaction between beads of different type can be obtained from the Lafitte combination rules[S6]. In these there is one free parameter $k_{ij}$, which as a reasonable starting point may be set to zero. When possible, the value of $k_{ij}$ should be tuned by comparing the simulation results to experimental values for a macroscopic property, such as solubilities, interfacial tension, vapour-liquid compositions, etc.

## Performance testing

In order to test the application performance under load we employ the Locust framework, which is written in Python and can simulate millions of users accessing the application. The

testing was done over gigabit Ethernet to avoid the network affecting the result.

To use Locust we write a small Python script that defines the tasks that should be tested. For testing Bottled SAFT, we ran several tasks with different weights to approximate real world load. These tasks included visiting the front page, searching for a random CAS number from a list of 2000 valid ones, running the correlation with custom input, etc. Initial testing indicated that the application running on a single CPU core could serve $\sim$ 7000 database lookups per hour and $\sim$ 70 000 visits to the front page per hour. To increase the lookup speed, profile-guided optimisation was performed of the search routines in the `search()` function, and a $\sim 10\times$ speedup was obtained for the various Pandas function calls by using NumPy arrays instead of Pandas dataframes as input to these lookup calls. To speed up displaying the front page, the web app runs behind nginx and uWSGI instead of Flask's built in server, which increases the speed by an order of magnitude. In the end, the application can sustain $\sim$ 70 000 database lookups per hour and $\sim$ 800 000 requests per hour for the front page. This was deemed sufficient. Further increases to the lookup speed could be obtained by using more than a single CPU core. Further increases to the front page loading speed could be obtained by rendering its template to a static HTML page and serving this with only nginx.

## Details concerning the ternary liquid-liquid system

The models for octane and benzene in Bottled SAFT are the same as those given in Ref. S7, where improved values for the liquid density at $T_r = 0.7$, $\rho_{0.7}$, have already been used in the correlation. The octane model has three beads per molecule, and the benzene model has two beads per molecule. For sulfolane, we searched for it in Bottled SAFT, a result was found, and the refined value for the liquid density $\rho_{0.7} = 8343.7$ mol/m$^3$ was entered. The three bead sulfolane model given as a final result by Bottled SAFT was selected, and the raaSAFT script given was downloaded and used, together with the octane and benzene models already found in raaSAFT.

The cross-interaction between the two main components of each liquid phase, sulfolane and octane, was tuned to match approximately the mutual solubilites found in experiments performed at 373.15 K[S8]. The tuning gave $k_{ij} = 0.07$ for use in the Lafitte combination rules.

Density profiles were computed from the simulation. These are shown in figure S1, together with a snapshot from the simulation showing a side view of the system.

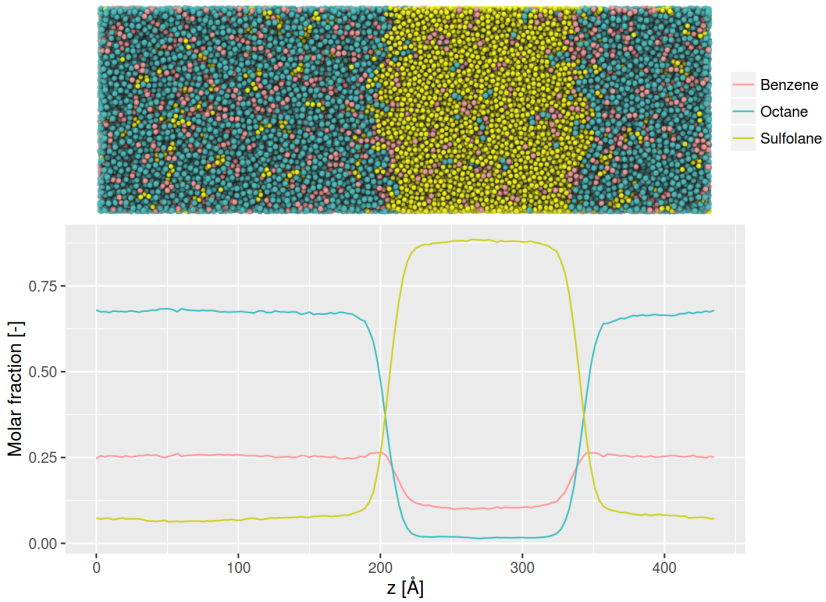



Figure S1: Top: side view from the molecular simulation. Bottom: molar fractions plotted along the elongated direction of the simulation box. The sulfolane-rich phase is located in the middle, and on both sides the octane-rich phase is found. Note that periodic boundary conditions are used.

# References

(S1) Müller, E. A.; Jackson, G. Force-Field Parameters from the SAFT-$\gamma$ Equation of State for Use in Coarse-Grained Molecular Simulations. *Annual review of chemical and biomolecular engineering* **2014**, *5*, 405–427.

(S2) Ramrattan, N.; Avendaño, C.; Müller, E.; Galindo, A. A corresponding-states framework

for the description of the Mie family of intermolecular potentials. *Molecular Physics* **2015**, *113*, 932–947.

(S3) Yaws, C. L. *Thermophysical properties of chemicals and hydrocarbons*; William Andrew: Norwich, NY, 2008.

(S4) McKinney, W. Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference. 2010; pp 51 – 56.

(S5) McKinney, W. Python Data Analysis Library pandas: Python Data Analysis Library. http://pandas.pydata.org/.

(S6) Avendaño, C.; Lafitte, T.; Adjiman, C. S.; Galindo, A.; Müller, E. A.; Jackson, G. SAFT-$\gamma$ force field for the simulation of molecular fluids: 2. Coarse-grained models of greenhouse gases, refrigerants, and long alkanes. *The Journal of Physical Chemistry B* **2013**, *117*, 2717–2733.

(S7) Herdes, C.; Totton, T. S.; Müller, E. A. Coarse grained force field for the molecular simulation of natural gases and condensates. *Fluid Phase Equilibria* **2015**, *406*, 91–100.

(S8) Ko, M.; Im, J.; Sung, J. Y.; Kim, H. Liquid-Liquid Equilibria for the Binary Systems of Sulfolane with Alkanes. *Journal of Chemical & Engineering Data* **2007**, *52*, 1464–1467.

# Paper 8

# A multiscale method for simulating fluid interfaces covered with large molecules such as asphaltenes

Åsmund Ervik[a,*], Morten Olsen Lysgaard[b], Carmelo Herdes[d], Guadalupe Jiménez-Serratos[c], Erich A. Müller[c], Svend Tollak Munkejord[e], Bernhard Müller[a]

[a] *Department of Energy and Process Engineering, NTNU, NO-7491 Trondheim, Norway*
[b] *Department of Mathematics, NTNU, NO-7491 Trondheim, Norway*
[c] *Department of Chemical Engineering, Imperial College London, London SW7 2AZ, United Kingdom*
[d] *Department of Chemical Engineering, University of Bath, Claverton Down, Bath, Somerset BA2 7AY, United Kingdom*
[e] *SINTEF Energy Research, P.O. Box 4761 Sluppen, NO-7465 Trondheim, Norway*

**Abstract**

The interface between two liquids is fully described by the interfacial tension only for very pure liquids. In most cases the system also contains surfactant molecules which modify the interfacial tension according to their concentration at the interface. This has been widely studied over the years, and interesting phenomena arise, e.g. the Marangoni effect. An even more complicated situation arises for complex fluids like crude oil, where large molecules such as asphaltenes migrate to the interface and give rise to further phenomena not seen in surfactant-contaminated systems. An example of this is the "crumpling drop" experiments, where the interface of a drop being deflated becomes non-smooth at some point. In this paper we report on the development of a multiscale method for simulating such complex liquid-liquid systems. We consider simulations where water drops covered with asphaltenes are deflated, and reproduce the crumpling observed in experiments. The method on the nanoscale is based on using coarse-grained molecular dynamics simulations of the interface, with an accurate model for the asphaltene molecules. This enables the calculation of interfacial properties. These properties are then used in the macroscale simulation, which is performed with a two-phase incompressible flow solver using a novel hybrid level-set/ghost-fluid/immersed-boundary method for taking the complex interface behaviour into account. We validate both the nano- and macroscale methods. Results are presented from nano- and macroscale simulations which showcase some of the interesting behaviour caused by asphaltenes affecting the interface. The molecular simulations presented here are the first in the literature to obtain the correct interfacial orientation of asphaltenes. Results from the macroscale simulations present a new physical explanation of the crumpled drop phenomenon, while highlighting shortcomings in previous hypotheses.

*Keywords:*

## 1. Introduction

Interfacial tension is a remarkable phenomenon, in that the tumultuous interactions of fluid molecules of different types and shapes give rise to macroscopic interfaces being not only smooth and stable, but indeed well-described by a single material constant, viz. *the interfacial tension* $\gamma$. This holds true for an impressive number of fluid molecules that may be polar or non-polar and may have different topology and size. It is by adding a third phase to the system, this phase being interfacially active either by virtue of amphiphilicity or

*Corresponding author
*Email addresses:* asmunder@pvv.org (Åsmund Ervik), morten@lysgaard.no (Morten Olsen Lysgaard), c.e.herdes.moreno@bath.ac.uk (Carmelo Herdes), m.jimenez-serratos@imperial.ac.uk (Guadalupe Jiménez-Serratos), e.muller@imperial.ac.uk (Erich A. Müller), svend.t.munkejord@sintef.no (Svend Tollak Munkejord), bernhard.muller@ntnu.no (Bernhard Müller)

through being poorly soluble in either fluid, that more complicated dynamics arise. The simplest case is that of amphiphilic surfactant molecules, where Marangoni [1, 2] first described the effects of nonuniform interfacial tension, and Gibbs in his seminal treatise [3] was the first to consider the effective elasticity of the interface imparted by the surfactants. The first mention of the deliberate use of surfactants to alter the interfacial properties of liquids is, however, much earlier. In book two of Pliny the Elder's encyclopedic *Naturalis Historia* [4, 5] (77 AD) it is mentioned that divers would release small oil drops from their mouth, to smooth the surface of the sea and thus increase the amount of light transmitted down to them.

Even though surfactants give rise to richer dynamics, their effects can be modelled using simple equations dependent on macroscopic parameters, using the approaches developed by Gibbs [3], Pockels [6], Szyszkowski [7], Langmuir [8] and Frumkin [9] (among others); see Levich and Krylov [10] (or the book by Levich [11]) for a good introduction. Apart from studies using these approaches for systems with low surfactant concentration, there is also a rich field of study into the various phases and phase transitions in the three-component systems when the number of surfactant molecules becomes comparable to the number of fluid molecules. However, we will not discuss this in further detail here. In the end, the effect of surfactants at low concentration is an interfacial tension which may vary along the interface, according to the variation in interfacial concentration of the surfactant. The relation between concentration and interfacial tension is given by e.g. a Langmuir equation of state.

In more recent years, as experimental techniques have increased in sophistication and soft matter has come to be a field of its own, attention has also turned to larger molecules at fluid interfaces. These larger molecules may originate from biological systems, being e.g. proteins or lipids, or they may originate from complex fluids such as crude oil, for instance asphaltenes. These molecules cause effects beyond those seen in surfactant systems, e.g. the crumpling upon deflation of red blood cells [12] or asphaltene-covered drops [13, 14]. It follows that something more than a (possibly varying) interfacial tension appears in these systems. We shall focus our attention here on asphaltenes, and will use this term rather than "large molecules" for the remainder of the paper, but the method remains general.

Let us then consider the asphaltenes. Having earned a reputation as the "cholesterol of crude oil", this component causes many problems in petroleum extraction, processing and refinement. As the name suggests, asphaltenes are similar in appearance to asphalt, and are found in large quantities in bitumen, but are present to some degree in most crude oils. The first reported discussion of asphaltenes is by Boussingault [15] who coined the term in 1836 (see Hoepfner [16, Chapter 1.1] for a historical review) but there is still today disagreement about their properties. Indeed, there was debate in the literature [17] up until recently on what their average molecular weight is. Furthermore, the precise definition of what constitutes an asphaltene is still not agreed upon. A commonly used operational definition is that it is the part of crude oil which is soluble in toluene but insoluble in $n$-heptane, as codified e.g. in the ASTM D 6560 – 00 standard. (From here on, we will denote $n$-heptane simply by heptane; branched alkanes are not considered in this work.) But this definition in terms of solubilities is more a description of how to isolate asphaltenes in the laboratory, as opposed to a definition of what they are.

The case is not clearer from the molecular perspective, as advanced experimental techniques such as neutron scattering or high-resolution mass spectrometry [18] have shown that there are thousands of different empirical formulae in a given asphaltene sample. Samples from different oil fields around the world have different asphaltene compositions. Furthermore, since asphaltenes have of the order of 50 carbon atoms, even for a single empirical formula the number of isomers is in the trillions. One may compare asphaltene molecules to snowflakes; no two are exactly the same. It has been argued that asphaltenes are among the most complex materials ever studied [16].

The effect of asphaltenes on liquid-liquid interfaces is also complicated. One such effect is that they make it hard to separate emulsions of water and crude oil. See e.g. Jones et al. [19], Gafonova and Yarranton [20], Sjöblom [21], Kokal [22], Kilpatrick [23] for reviews on water-in-crude oil emulsion stability. Emulsion stability has been directly linked to the properties that the asphaltenes impart on the interface [24, 25]. The asphaltenes give rise to interesting phenomena such as the previously mentioned crumpled drops reported by Yeung et al. [13], Pauchard et al. [14].

In the present paper we report on our development of a multiscale method that enables the simulation of liquid-liquid interfaces covered with asphaltenes (or other large molecules). This multiscale method is loosely

coupled, with equilibrium simulations at the nanoscale providing parameters for dynamic simulations at the macroscale. Many classifications exist of multiscale methods, some of which have a much tighter coupling between the scales than in the present approach, see e.g. [26, 27] for reviews of multiscale methods.

The outline of this paper is as follows. In Section 2 we review the theory and present the methods used on the nanoscale and the macroscale, in particular the SAFT-$\gamma$ Mie approach to coarse-grained molecular dynamics simulation (Section 2.1), and the hybrid level-set/ghost-fluid/immersed-boundary method developed in this work for simulation of two-phase flows with complex interfaces(Section 2.4). The latter method is summarised in Section 3. Subsequently, in Section 4, we validate these methods using standard test cases for interface-capturing methods (Section 4.1), and by comparing the predictions from molecular simulations to experimental results (Section 4.2). We then present the results obtained with the method in Section 5, discuss the implications of these results in Section 6, and finally give some concluding remarks in Section 7.

## 2. Theory and methods

### *2.1. Nanoscale: theory*

The complicated behaviour of liquid-liquid interfaces contaminated with asphaltenes is caused by the interactions between the molecules at the interface. It is thus tempting to try and explain the interfacial phenomena by modelling the molecular interactions. These are, in turn, also complicated, and one is forced to make simplifications. On the most basic level, many-body quantum mechanics is what lies behind the nature and interactions of molecules. Fortunately, on the scale of interactions between large molecules, one may develop an effective theory which is much simpler. This theory has roots going back to the dawn of thermodynamics, when pioneering efforts were made to understand how the microscopic nature of fluids could explain their behaviour. The works by Maxwell, Boltzmann, van der Waals, Lennard-Jones, Mie, Chapman, Engskog and others paved the way to our understanding of simple fluids from the molecular perspective; see e.g. Chapman et al. [28] for an overview. For less simple fluids, it was not until the advent of computer simulations that the molecular perspective was able to provide some insights.

One of the major challenges in simulating the behaviour of chemicals comes from the need for good models that can accurately predict physical and chemical properties. For many simple substances, experimental data may already be available or easy to obtain. However, many interesting systems contain one or several chemically complex species, e.g. polymers, surfactants, and our asphaltenes. The quantitative prediction of the thermodynamic properties of such systems, especially their phase behaviour and mesoscopic structure, is very challenging. Simple equations of state are typically unable to make good predictions of the properties of structured fluids. As this much is clear, one must resort to computer simulations of these more complex fluids.

Many molecular simulation methods rely on force fields or other empirical parameters that are fitted to reproduce the properties of particular classes of compounds, or to a particular large data set. These methods may have trouble predicting the properties of complex mixtures. Truly *ab initio* prediction, using electronic structure methods, is possible only for very small systems and processes spanning very short time scales, and cannot be used directly to predict the properties of structured fluids or complex materials. It is possible to construct models for complex systems by first building models for the different (smaller) component parts using more predictive methods, and then eliminating the unimportant degrees of freedom.

This leads to a coarse-grained description that can be used to predict properties of more complex materials. However, it is time-consuming to generate the data for the smaller building blocks, and it is often not clear what the best way is when coarse-graining out the smaller/faster degrees of freedom. An alternative approach is to construct a coarse-grained model directly using experimental data for the building blocks ("top-down" coarse graining) – in this case the evaluation of the residual when performing the fit depends on a relatively costly simulation, making the cost of fitting very high.

Recently, however, an alternative approach has been successfully used to fit coarse-grained models to thermodynamic data without using simulations, but rather using statistical mechanical perturbation theory. This novel approach, the SAFT-$\gamma$ method, allows for the construction of coarse-grained models suitable for molecular simulation of complex fluids using available experimental data for the constituent blocks. The

SAFT equation of state is a perturbation approach based on a well-defined Hamiltonian. The reader is referred to several reviews on SAFT that describe the various stages of its development, up to the current SAFT-$\gamma$ Mie approach [29, 30, 31, 32, 33, 34]. In the current approach, a Mie [35] potential gives the forces between the coarse grained *beads* in the method,

$$V(r) = C(n,m)\,\epsilon\left[\left(\frac{\sigma}{r}\right)^{n} - \left(\frac{\sigma}{r}\right)^{m}\right], \tag{1}$$

$$C(n,m) = \left(\frac{n}{n-m}\right)\left(\frac{n}{m}\right)^{m/(n-m)}, \tag{2}$$

where $r$ is the distance between a pair of beads, and $\epsilon$ and $\sigma$ are the adjustable parameters relating to the energy and distance scales. Each bead typically corresponds to 2–4 atoms heavier than hydrogen, together with the hydrogen atoms attached to the heavier atoms. Referring to Figure 1, the parameter $\epsilon$ corresponds to the depth of the potential well, and $\sigma$ to the distance $r$ where the potential switches from being repulsive to being attractive. Thus $\sigma$ is taken to correspond to the bead "diameter" for visualisation purposes. Note that in the $n \to \infty$ limit, $V(r)$ becomes a hard-sphere potential with diameter equal to $\sigma$.

It is important to note that in the present approach, the attractive exponent $m$ is fixed at the value of six, but the short-range repulsion exponent $n$ takes on different values reflecting the average softness or hardness of the potential for a given molecule. The fact that $m = 6$ is fixed is due to the observation that these two exponents are not independent parameters [36], so fixing one simplifies the parameter space considered. The effect of varying $n$ is illustrated in Figure 1 where the potential is plotted for $n$ ranging from 8 to 24 in steps of 2. The $n = 12$ or Lennard-Jones potential is shown in a stronger orange colour. It is seen that allowing $n$ to vary allows for using softer or harder potentials for beads representing different molecules, which is something that cannot be achieved in many other approaches which rely on only the Lennard-Jones potential.

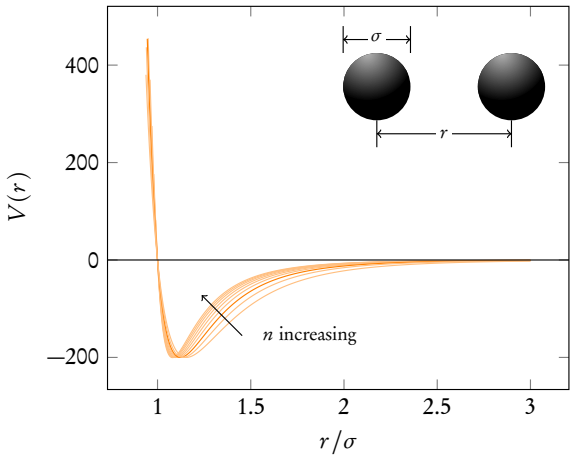



Figure 1: The Mie (m,6) potential shown for $\epsilon = 200$ (in arbitrary units) and $n$ varying from 8 to 24, as a function of the dimensionless distance $r/\sigma$. The Lennard Jones (12,6) potential is shown in a stronger colour. In the corner, the two beads are illustrated with a distance $r$ and a "diameter" $\sigma$.

There are several methodologies proposed in the literature to obtain parameters for coarse grained models [37] . Most common approaches start from a fine-detailed model, usually an atomistically-detailed characterization and integrate out degrees of freedom which are deemed unessential [38]. This procedure, known as a bottom-up approach, inevitably discards information and produces potentials which can rarely be used in states points other than those used to develop them.

A pathway to circumvent these problems is to employ a top-down or thermodynamic approach, where the force field of the coarse grained sectors is an effective or average potential capable of reproducing

macroscopic thermodynamic properties. These top-down coarse grained models provide by their own nature parametrizations which are usually robust, representative or transferrable. Two notable examples are the MARTINI force field [39, 40], which employs a group contribution approach targeted at biomolecular simulations where uniform-sized coarse grained beads have been fitted to water/octanol solubilities.

In a group contribution approach, molecules which have different parts with different properties are modelled using combinations of beads which represent each individual part. As an example, octanol could be modelled with one bead that represents the head with the alcohol group, based on the thermophysical properties of ethanol, and the other beads which represent the aliphatic tail could be based on the properties of hexane.

A more refined coarse-grained model is the SAFT-$\gamma$ force field [41], which employs an analytical equation of state as the bridge between macroscopic thermophysical properties and the underlying intermolecular potential that can effectively generate them. The most direct method of parametrization of coarse grained segments and chemical moieties is to use the appropriate version of the SAFT equation of state to fit experimental phase equilibria data, e.g. saturated liquid densities, vapour pressures, etc. essentially equating the free energy of a coarse grained model to that obtained from the analysis of experimental data [42, 43, 33, 44].

However, if one recognizes the conformal nature of the underlying Mie potential, one can formulate the equation of state in terms of a corresponding-states model and can express the properties of any non-associating fluid in terms of a finite set of defining properties: a critical temperature, the acentric factor and a well-defined density. This approach [45] greatly simplifies the parameter estimation without detriment to the robustness of the methodology, as exemplified by the predictions of adsorption [46], transport and interfacial properties [41] which are not part of the original fit, and the description of complex molecules such as surfactants [47, 48], resins and asphaltenes [49]. This approach is used to construct the force field parameters used in this work. The parameters for different coarse-grained beads are given in Appendix B.

### *2.2. Nanoscale: numerical methods*

Once the parameters for the intermolecular force field have been established for all molecules under consideration, one can proceed to study the system. Since only the two-bead problem has an analytical solution, a numerical approach is required for any realistic system. There are two fundamental approaches, namely Monte Carlo methods and Molecular Dynamics (MD) methods. The ergodic hypothesis asserts that these two are the same, i.e. that ensemble averages and time averages give the same answers. Molecular dynamics is the approach used in this work, so we will not discuss Monte Carlo methods in any detail.

In molecular dynamics, the equations of motion are solved to evolve the system in time from some initial state. The equations of motion are Newton's second law for each of the $N$ beads in the system, with the force given by the Mie potential in the approach used here. Denote by $\mathbf{x}_i$ the position of bead $i$ with mass $m_i$. The equations of motion are then

$$\frac{\partial^2 \mathbf{x}_i}{\partial t^2} = \sum_{j \neq i} \frac{F(|\mathbf{x}_j - \mathbf{x}_i|)}{m_i} \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|} \tag{3}$$

where the forces are assumed to be conservative, and depending only on the distance between two beads. Thus the forces are given by a potential $V(r)$. This must be specified; a Mie potential is used here, as discussed previously. This potential is short-ranged, so a *cutoff* $r_{\text{cut}}$ is specified, beyond which $V(r) = 0$. The equations of motion are solved numerically using a symplectic integration method, since this ensures the simulation is stable over long times and that the energy drift is very small. An example of such a method is the velocity Verlet method. Here, the velocities $\mathbf{v}_i$ and positions $\mathbf{x}_i$ are stored at each time step $n$. The

system is integrated forward in time to the time step $n+1$, with a step length $\Delta t$, according to

$$\mathbf{v}_i^{n+1/2} = \mathbf{v}_i^n + \sum_{j \in \mathbb{N}(i,n)} \frac{\Delta t}{2m} F(|\mathbf{x}_j^n - \mathbf{x}_i^n|) \frac{\mathbf{x}_j^n - \mathbf{x}_i^n}{|\mathbf{x}_j^n - \mathbf{x}_i^n|} \tag{4}$$

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \mathbf{v}_i^{n+1/2} \Delta t \tag{5}$$

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^{n+1/2} + \sum_{j \in \mathbb{N}(i,n+1)} \frac{\Delta t}{2m} F(|\mathbf{x}_j^{n+1} - \mathbf{x}_i^{n+1}|) \frac{\mathbf{x}_j^{n+1} - \mathbf{x}_i^{n+1}}{|\mathbf{x}_j^{n+1} - \mathbf{x}_i^n|} \tag{6}$$

Here $\mathbb{N}(i,n)$ is the neighbourlist of bead $i$, i.e. the indices of all beads which are within the cutoff distance of bead $i$ at the time step $n$. Using a neighbourlist dramatically speeds up the algorithm, but comes at a storage cost; this data structure accounts for the bulk of the memory used by a molecular dynamics code. The fact that all interactions are local makes the method well-suited for parallelisation, both on classical CPUs and on accelerators such as graphical processing units (GPUs).

The beads are contained in a virtual simulation box, typically with periodic boundary conditions. The initial state must be generated somehow. In the present work we start from random initial conditions at a low density. To go from this initial state to the desired system state, e.g. a temperature $T$ and a pressure $p$, the system is evolved in the isothermal-isobaric or $NpT$ ensemble, where a thermostat and a barostat are employed to adjust the bead velocities and the simulation box size, respectively, to obtain the desired system state. Once this state is reached, the simulation box size is fixed and the system is subsequently evolved in the canonical or $NVT$ ensemble. The reader is referred to standard textbooks on molecular dynamics, e.g. by Allen and Tildesley [50], Frenkel and Smit [51], for further details.

While the first molecular dynamics simulations [52, 53] were limited to two-dimensional systems and simple potentials, the results obtained provided fascinating new insights into the molecular world. With the exponential increase in computing power since the 1950's, molecular dynamics simulations today have probed systems with hundreds of billions of atoms [54], or entire virus capsules [55], using large high-performance computing systems.

Combining a coarse-grained approach to molecular dynamics, such as the SAFT-$\gamma$ Mie force field used here, with the power of general purpose GPUs, we obtain speedups of more than three orders of magnitude over atomistically detailed simulations running on usual CPUs [56]. This means the present simulations, even though they evolve systems with about one million atoms for hundreds of millions of timesteps in total, are run on computational resources that can fit inside a desktop computer. The simulations reported here are run in parallel typically on two or four GPUs; separate simulations are run simultaneously on different nodes of a GPU cluster, consuming in total 8000 GPU-hours during this study.

For coarse-grained molecular dynamics simulations we employ our raaSAFT code [56], a framework for simulations using the SAFT-$\gamma$ Mie force field. raaSAFT leverages existing molecular dynamics codes to do the heavy lifting. Here HOOMD-blue [57, 58] is used, a modern GPU-first code that shows excellent performance and scalability on multiple GPUs.

HOOMD-blue takes a conservative approach to molecular dynamics, and uses algorithms that do not sacrifice accuracy for speed. The simulations are run in the isothermal-isobaric ($NpT$ ) or the isothermal-isochoric ($NVT$ ) ensemble, and the system is evolved in these ensembles using the Martyna-Tobias-Klein approach [59]. This approach gives dynamics that are provably time-reversible and energy-preserving. In the light of this, it is interesting how the simulations show time-irreversible results, such as the clustering of asphaltenes. This is a variant of Loschmidt's paradox. Recent work by Hoover et al. [60, 61, 62] might provide some insight into the explanation of this, but so far it appears to remain an open question.

When systems with immiscible fluids are considered, it is of interest to compute the interfacial tension. To do this, the simulation box is elongated in one direction, and this asymmetry causes the formation of interfaces along the two shorter dimensions of the box, since this minimises the free energy of the interfaces. Note that since periodic boundary conditions are employed, it is a topological impossibility to have an odd number of interfaces between two liquids, so the desired system state has two slabs of liquid which are in contact at two interfaces.

For this system, we may compute the interfacial tension from the integral of the anisotropy in the diagonal elements of the stress tensor, $\sigma_{ii}$, along the elongated box dimension. To be precise, assuming the box is elongated in the $z$-direction where the box dimension is $L_z$, the interfacial tension is given by

$$\gamma = \frac{L_z}{2} \int_0^{L_z} \left( -\sigma_{zz} + \frac{1}{2}(\sigma_{xx} + \sigma_{yy}) \right) dz. \tag{7}$$

This is referred to as the mechanical route to the interfacial tension, and goes back to Kirkwood and Buff [63]. For a comparison of this with alternative methods, see e.g. [64]. Note that the integral here can be split into three parts, so molecular dynamics software such as HOOMD-blue typically compute the values

$$p_{xx} = \int_0^{L_x} -\sigma_{xx} dx, \tag{8}$$

and similarly for $p_{yy}, p_{zz}$, and refer to these as the diagonal components of the "pressure tensor". From these three numbers we may compute the interfacial tension using Equation (7). Since molecular dynamics simulations are inherently noisy, in particular for properties related to the pressure, $\gamma$ computed from this expression for a single point in time will fluctuate significantly from one time step to the next. To obtain a reliable value for $\gamma$, time averages must be employed.

To also compute the elasticity of the interface, we follow refs. [65, 66], where the elasticity $K_a$ is computed from the change in interfacial tension $\Delta\gamma$ (as computed from Equation (7)) when the interfacial area is changed from $A_0$ to $A$, given by the expression

$$K_a = \frac{\Delta\gamma}{A/A_0 - 1}. \tag{9}$$

These two parameters, $\gamma$ and $K_a$, are subsequently employed as material parameters in the macroscale simulations.

### *2.3. Macroscale: theory*

The flow inside and around water drops in oil behaves according to the incompressible Navier-Stokes equations. For a derivation of these and a general overview, the reader is referred to standard textbooks e.g. by Batchelor [67] or Lamb [68]. The incompressible Navier-Stokes equations for single-phase flow read

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \boldsymbol{\nabla} \cdot (\mu \boldsymbol{\nabla} \mathbf{u}) + \rho \mathbf{f}, \tag{10}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{11}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \tag{12}$$

$$\mathbf{u}_{\partial\Omega}(t) = \mathbf{g}(t), \tag{13}$$

where $\partial\Omega$ is the domain boundary and $\mathbf{g}(t)$ is the velocity boundary condition. The initial condition is $\mathbf{u}_0$. The viscosity $\mu$ and the density $\rho$ are assumed to be constant throughout the domain. The velocity is denoted by $\mathbf{u}$ and the pressure by $p$, and $\rho\mathbf{f}$ is a body force such as gravity. This can be extended to handle two fluid phases, with different viscosities and densities. Let $\Omega_1$ and $\Omega_2$ denote the domains filled with fluid 1 and fluid 2, respectively and $\Gamma$ denote the interface separating the two fluids, i.e. we have $\Omega = \Omega_1 \cup \Omega_2$, as illustrated in Figure 2.

The tension on $\Gamma$ can be modelled as a contribution to Equation (10), localised at the interface. In this work we consider one-dimensional or axisymmetric two-dimensional interfaces, and in both cases the interface is parametrised as a one-dimensional curve $\mathbf{x}_I(s)$. In the axisymmetric case, this curve is swept around the azimuthal angle $\phi$ to form the two-dimensional interface. The interfacial tension has two contributions in the axisymmetric case, $T_s$ and $T_\phi$, both of which may vary as functions of the position $s$ along the interface. In the case of a one-dimensional interface, $T_\phi = 0$.

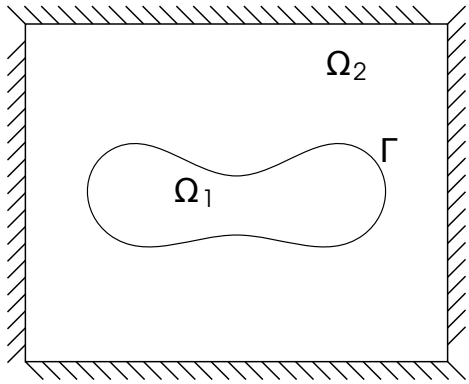



Figure 2: Two fluid phases, the inner fluid 1 in $\Omega_1$ and the outer fluid 2 in $\Omega_2$, and the interface $\Gamma$ separating them.

The interfacial force contribution to the body force in Equation (10) is then

$$\mathbf{f}_s(\mathbf{x},t) = \int_\Gamma \left( \frac{\partial T_s}{\partial s}\mathbf{t} + T_s\kappa_s\mathbf{n} + T_\phi\kappa_\phi\mathbf{n} \right) \delta(\mathbf{x}-\mathbf{x}_I(s))\mathrm{d}\mathbf{s}, \tag{14}$$

where $\frac{\partial T_s}{\partial s}$ is the derivative of the meridional tension along the interface, $\mathbf{t}$ is the interface tangent, $\kappa_s, \kappa_\phi$ are the curvatures, $\mathbf{n}$ is the principal unit normal vector, and $\delta$ is the Dirac delta function. We assume here that the interfacial curvature is small enough that the interface is approximately flat on the microscopic level. This is a good approximation for drops with radius $R \gg 1\text{nm}$ [69], which does not pose a significant restriction. The interfacial tension for pure fluids is then dependent just on the temperature and the pressure.

In Figure 3 we illustrate the tensions for the axisymmetric case, together with the two coordinate systems employed, on a spherical drop. Note that the drop is not restricted to being spherical, so none of the coordinate systems are spherical, and $s$ is an arc length, not an angle. The line parametrised by $s$, which represents the drop surface, actually lies in the $(r,z)$ plane, but in the figure the line is rotated out of the plane in order to avoid clutter. Since we assume axisymmetry, neither of the coordinate systems will have points discretising the $\phi$-direction in the numerical methods.

On the right-hand side of this figure, the tensions $T_s$ and $T_\phi$ are indicated as line segments in the directions of strain, using varying red and green colours, respectively. Both tensions may vary along the $s$-direction, but are constant in the $\phi$-direction since axisymmetry is assumed. As is indicated in Equation (30) in the next section, only the variation in $T_s$ can give rise to a tangential force. From the figure we understand this, since any non-normal component of the force caused by $T_\phi$ would have to be either in the azimuthal direction, violating the assumption of axisymmetry; or in the meridional direction, but this force would be binormal to the strain line, which is not possible.

The description of an interface between two fluids can be as simple as a constant interfacial tension $T_s = T_\phi = \gamma$, or more complicated due to molecules which are interfacially active. In any case, the formation of the interface gives an increase in the energy of the system, and we denote the energy density of the interface by $w$. It should be noted that while the interfacial energy (in $\mathrm{J/m^2}$) and the interfacial tension (in $\mathrm{N/m}$) are identical for the case of simple fluids, this is *not* the case for an interface with a more complicated interface which has elastic properties; see e.g. [70] for details.

Following [71], we write $w$ as a function[1] of the interfacial deformations $\lambda_s$ and $\lambda_\phi$,

$$w = w(\lambda_s, \lambda_\phi). \tag{15}$$

[1] $w$ is technically a functional; see e.g. [72] for a more mathematical formalism.

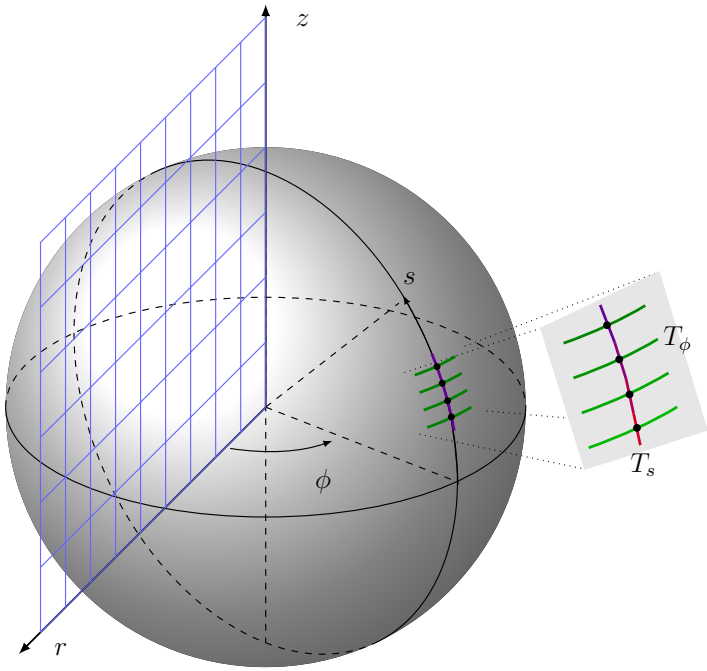



Figure 3: Illustration of the two coordinate systems employed, and the tensions on the drop. The blue grid illustrates the $(r, z)$ coordinate system, where a structured grid is employed and the Navier-Stokes equations are solved. The grey sphere illustrates the shape of a drop. On this drop, the coordinates $s, \phi$ are used. Note that $s$ is an arc length, not an angle. On the right, the tensions are illustrated with red ($T_s$) and green ($T_\phi$) line segments. The varying colours indicate varying tensions. Note that while both tensions vary from point to point, only $T_s$ varies in the direction parallel to the corresponding strain $\lambda_s$, while $T_\phi$ is constant in the direction parallel to $\lambda_\phi$ (due to axisymmetry).

These deformations refer to the length of an interfacial element $l_i$ at time $t$ relative to the undeformed length at $t = 0$, i.e. $\lambda_s = l_i(t)/l_i(0)$. The interface will tend to deform such that the energy is minimised, while maintaining a constant volume inside. The minimal energy shape depends on the form of $w$; for the familiar case of constant interfacial tension the minimal energy shape is a sphere.

We proceed to derive the general tensions for an interface which is described by a constant interfacial tension $\gamma$ and a Hookean elasticity $K_a$, following again [71]. The bending rigidity of the interface is assumed to be zero; a non-zero bending rigidity may be considered in future work. As noted previously, we parametrise the interface using the coordinate $s$ along the interface in the $(r, z)$ plane. All quantities in the system are constant along the azimuthal direction $\phi$. Under these assumptions the energy $w$ is given by

$$w = \frac{1}{2}\left[\frac{K_a}{1-\nu^2}\left((\lambda_s-1)^2 + 2\nu(\lambda_s-1)(\lambda_\phi-1) + (\lambda_\phi-1)^2\right) + \gamma\lambda_s\lambda_\phi\right]. \tag{16}$$

To obtain the tensions one takes the partial derivatives of the energy with respect to the deformations; to be precise,

$$T_s = \frac{1}{\lambda_\phi}\frac{\partial w}{\partial \lambda_s} = \frac{K_a}{\lambda_\phi(1-\nu^2)}\left(\lambda_s + \nu\lambda_\phi - (1+\nu)\right) + \gamma, \tag{17}$$

$$T_\phi = \frac{1}{\lambda_s}\frac{\partial w}{\partial \lambda_\phi} = \frac{K_a}{\lambda_s(1-\nu^2)}\left(\lambda_\phi + \nu\lambda_s - (1+\nu)\right) + \gamma, \tag{18}$$

$$\tag{19}$$

which are the tensions in the meridional and azimuthal directions, respectively. These are inserted into Equation (14) to obtain the force on the interface. In these expressions $\nu$ denotes the Poisson ratio, which is a material constant that couples the meridional and azimuthal deformations. In the present work, $\nu = 0.3$ is used, which is a reasonable assumption for elastic interfaces such as those considered here [73].

### *2.4. Macroscale: numerical methods*

Having described the equations governing the system, we now consider how to solve these numerically. For the numerical methods, the equations must be discretised in space and time. In space, a structured, uniform and staggered grid is employed, and the derivatives are discretised using standard second-order finite differences, except for the convective term in Equation (10) which is discretised using the fifth-order WENO scheme [74, 75]. Due to the coupling between pressure and velocity, the Navier-Stokes equations are not a regular set of PDEs, but technically a differential-algebraic equation with an index-2 constraint given by the incompressibility equation. To solve this, we employ the pressure projection method due to Chorin [76], which leads to a splitting scheme for the time integration. In this scheme, we solve first for an intermediate (non-divergence-free) velocity field using an Euler step, then solve a Poisson equation for the pressure based on this intermediate velocity, and finally use the computed pressure to project the velocity field into the space of divergence-free velocity fields. The pressure Poisson equation takes up the bulk of the computation time, and much work has gone into developing fast numerical methods for this equation. In the present work we employ the BoomerAMG [77] preconditioned BiCGStab [78] method, through the Hypre [79] and PETSc [80] libraries, respectively. To obtain a larger stability domain than with only an explicit Euler step, several Euler steps are combined to form a Runge-Kutta step (following the approach by Kang et al. [81]), specifically the SSP-RK(2,2) method is employed here (using the notation of Gottlieb et al. [82]). The method does, however, remain first-order in time due to the irreducible splitting error introduced by the projection step; see [83] for a review of error reduction and of higher-order projection methods for the Navier-Stokes equations.

This summarises how the single-phase Navier-Stokes equations are solved. To extend this to two-phase flow, several methods are available. In previous work we have employed the combination of the level-set and ghost-fluid methods, which gives results that agree well with theory and experiments [84, 85, 86, 87, 88, 89], and which can handle topological changes in the interface, e.g. during drop coalescence. This method can handle a varying interfacial tension, and has been coupled with the Langmuir equation of state to simulate the effects of insoluble surfactants [86, 89].

To simulate interfaces with tensions that include a Hookean elasticity, this scheme had to be extended, as we will discuss in the following. To this end, a hybrid level-set/ghost-fluid/immersed-boundary method has been developed. To describe the hybrid method, we first give a brief overview of each of the methods it is constructed from. The development of the hybrid method is documented in greater detail in the MSc thesis of Lysgaard [90] for the two-dimensional case; the method was extended to axisymmetry later.

#### *2.4.1. The level-set method*

To solve the Navier-Stokes equations for two-phase flow, knowledge of the interface is required. For this, a popular choice is the level-set method, originally proposed by Osher and Sethian [91]. With this method the interface is encoded as a signed scalar distance field

$$\varphi(\mathbf{x}, t) = \begin{cases} \min_{\mathbf{x}' \in \Gamma(t)} \|\mathbf{x} - \mathbf{x}'\| & \text{if } \mathbf{x} \in \Omega_1 \\ -\min_{\mathbf{x}' \in \Gamma(t)} \|\mathbf{x} - \mathbf{x}'\| & \text{if } \mathbf{x} \in \Omega_2 \end{cases} \tag{20}$$

This gives an implicit definition of the interface,

$$\Gamma(t) = \{\mathbf{x} \in \Omega \mid \varphi(\mathbf{x}, t) = 0\}. \tag{21}$$

The interface moves according to the flow of the fluids. This advection is performed directly with the function $\phi$, and for this reason the level-set method is referred to as an implicit interface capturing method. The advection equation is then

$$\frac{\partial \varphi}{\partial t} + \hat{\mathbf{u}} \cdot \nabla \varphi = 0, \tag{22}$$

where $\hat{\mathbf{u}}$ is the fluid velocity field at the interface, extrapolated to the whole domain (as suggested by Adalsteinsson and Sethian [92]), which can be found by solving

$$\frac{\partial \hat{\mathbf{u}}}{\partial \tau} + S(\varphi)\mathbf{n} \cdot \nabla \hat{\mathbf{u}} = 0, \quad \hat{\mathbf{u}}|_{\tau=0} = u, \tag{23}$$

Here $\tau$ is a pseudo time, and $S$ is a smeared sign function which is zero at the interface, $S(\varphi) = \varphi/\sqrt{\varphi^2 + 2\Delta^2}$. We assume here that the Eulerian grid spacings are equal and denote these by $\Delta_x = \Delta_y = \Delta$. This equation is in principle solved to steady state, i.e. $\tau \to \infty$. In a recent paper, Sabelnikov et al. [93] presented an alternative approach which appears promising, since it has a lower computational cost.

As the level-set field is advected by Equation (22) it will become distorted and lose its signed distance-property. Because of this, the level-set function is reinitialised at regular intervals by solving

$$\frac{\partial \varphi}{\partial \tau} + S(\varphi_0)(|\nabla \varphi| - 1) = 0, \tag{24}$$

$$\varphi(\mathbf{x}, 0) = \varphi_0(\mathbf{x}), \tag{25}$$

to steady state [94, (7.4)]. Even though Equation (23) and Equation (24) are defined for the whole domain, we are only interested in the extrapolated velocity and the reinitialised field in a neighbourhood around the interface.

Interestingly the characteristics of both Equation (22), Equation (23) and Equation (24) originate at the interface and point outwards. This implies that if one solves the equations not for $\tau \to \infty$, but rather for $\tau \to N\Delta$, $N$ e.g. equal to 3, it gives a level-set function which is correct in a narrow band of width $3\Delta$ around the interface. This significantly reduces the computational cost of the method; see e.g. [92] for further details.

The properties required to calculate forces coming from a fluid interface are the interface normal vectors and curvature. Both of these are computed directly from $\varphi$,

$$\mathbf{n} = \frac{\nabla \varphi}{|\nabla \varphi|}, \quad \kappa = \nabla \cdot \mathbf{n}. \tag{26}$$

*2.4.2. The ghost-fluid method*

Given an interface-capturing method such as the level-set method, yet another method is required to impose the difference in material properties and the interfacial tension. Different methods are available, and one distinguishes between sharp-interface and smeared-interface methods. A sharp-interface method (such as the ghost-fluid [95] or immersed-interface methods [96]) is more accurate, but also more difficult to implement, as compared to a smeared-interface method (such as the continuum-surface-force [97] or immersed-boundary [98] methods).

With a smeared-out method, a mollified delta-function is used to spread a singular force out to several grid cells. With such an approach, the normal finite-difference approximations to derivatives can be used as there are no discontinuities in the solution, but rather very steep, smooth transitions.

By contrast, with the ghost-fluid method as used in this work, the discontinuities are incorporated directly into the numerical stencils. This means that there is an actual jump in the solution, and jump conditions are used to relate the values across the interface. For the case of two-phase flow with a constant interfacial tension, the jumps are given by

$$[\![\mathbf{u}]\!] = 0, \tag{27}$$

$$[\![p]\!] = 2[\![\mu]\!]\mathbf{n} \cdot \boldsymbol{\nabla}\mathbf{u} \cdot \mathbf{n} - \gamma\kappa, \tag{28}$$

$$\begin{aligned}[\![\mu \boldsymbol{\nabla}\mathbf{u}]\!] = [\![\mu]\!]\Big( &(\mathbf{n} \cdot \boldsymbol{\nabla}\mathbf{u} \cdot \mathbf{n})\mathbf{nn} + (\mathbf{n} \cdot \boldsymbol{\nabla}\mathbf{u} \cdot \mathbf{t})\mathbf{nt} \\ &- (\mathbf{n} \cdot \boldsymbol{\nabla}\mathbf{u} \cdot \mathbf{t})\mathbf{tn} + (\mathbf{t} \cdot \boldsymbol{\nabla}\mathbf{u} \cdot \mathbf{t})\mathbf{tt}\Big)\end{aligned} \tag{29}$$

Here $\mathbf{n}, \mathbf{t}$ are the normal and tangent vectors at the interface, and we denote tensors formed by the outer product as e.g. $\boldsymbol{\nabla}\mathbf{u}$. We take the normal vector to be pointing outwards on a drop, and then the jump

$[\![\cdot]\!]$ is the difference between the external and internal values, e.g. $[\![\mu]\!] = \mu_2 - \mu_1$. It should be noted that these expressions have been written in a form that gives faster code when implemented, see Lervåg [99] for a derivation and for a more thorough description of the ghost-fluid method.

In the hybrid method developed here, these equations are used without the term $\gamma\kappa$ in Equation (28), since the tension is then handled by the immersed-boundary method instead. The term is included when the regular level-set/ghost-fluid method is used as a reference for testing the hybrid method.

*2.4.3. Motivation for the hybrid method*

To compute the tensions using Equations (17) and (18) requires knowledge of the interfacial deformations $\lambda_s, \lambda_\phi$. We may prove that the level-set function, or any similar scalar marker function, does not contain the information required to compute this. Equivalently, the marker function does not contain information about compression or stretching of the interface. To have compression or stretching of an interface in incompressible flow, assuming no sources or sinks are present, the velocity component tangential to the interface has to be nonzero. By considering the projection operator

$$P_{\parallel}(\varphi) = (1 - \nabla\varphi(\nabla\varphi\cdot\,))$$

which projects $\mathbf{u}$ into the space of velocity fields that are tangential to the interface, one may easily prove, using the signed distance property of $\varphi$, that only the velocity component normal to the interface gives a non-zero contribution in the advection equation. In other words, the interface representation $\phi$ is invariant under velocity fields tangential to the interface. To store information about interfacial compression or stretching in an interface capturing method which uses an Eulerian marker function, one must resort to additional data structures to represent interfacial strain.

Alternatively, we may consider a hybrid interface-tracking method. Such methods have been successful at combining the best features of several methods, e.g. in the coupled level-set/volume-of-fluid (CLSVOF) method [100]. In the present work we have developed a hybrid level-set/ghost-fluid/immersed-boundary method. The immersed boundary method provides not only the required information about compression and stretching, but is also widely used and thoroughly tested with a general tension (i.e. elasticity and interfacial tension). Originally developed for simulating biological systems, e.g. blood flow through a heart [98], the immersed boundary method has been successfully applied to the simulation of red blood cells [12], which have similar properties to drops covered with elastic membranes.

Another important motivation for using the immersed boundary method is that it allows for refining the discretisation of the interface independently of the Eulerian grid. This increases the accuracy of the interfacial representation while the Eulerian grid remains the same, as indicated by the results of standard interface-capturing method test cases in Section 4.1. In particular for the crumpling drop case of interest here, this represents a large saving of computational cost, since crumpled interfaces like those discussed in Section 5, represented using the level-set function, would require at least an order of magnitude more Eulerian grid points than what is required to represent the flow field with sufficient accuracy. This would cause simulation times to be at least two orders of magnitude larger (one order of magnitude from increased cost in the Poisson solver, and one from the increased number of time steps required due to the stability condition). The simulations considered in Section 5 have runtimes of a few days running in serial; parallelisation would give some improvement in the time-to-solution, but even state-of-the-art solvers for the pressure Poisson equation show limited performance gains when the number of unknowns per CPU core is below $O(10000)$. Thus if one was to use an extended level-set/ghost-fluid method, having data structures to represent interfacial strain, and a parallelised code, the runtime for one of these cases would be of the order of months and the effort to implement the method would be much larger.

The reason for not completely switching to the immersed-boundary method is that the handling of density and viscosity differences across the interface is less accurate than with the level-set/ghost-fluid method; with the immersed-boundary approach the contributions from the viscosity difference in the jump conditions Equations (28) and (29) are typically not taken into account (see e.g. [101]), similar to when the level-set method is used together with the continuum-surface force method [97]. The reason for retaining the level-set method is that this eases the implementation, and that it may allow for simulations of drop coalescence in an

extended version of the hybrid method, since the level-set method handles changes in the interface topology very well.

*2.4.4. The immersed boundary method*

The key point of the immersed boundary method [72] is to allow solving the Navier-Stokes equations, or other continuum equations, on an Eulerian regular grid, while handling a large class of arbitrary deformable or rigid bodies embedded in the same domain. These bodies are described by Lagrangian coordinates. Thus a key element of the method is the transformations between the Lagrangian and Eulerian coordinates, and vice versa.

In addition to flexible interfaces as considered here, the immersed-boundary method can be used to simulate rigid bodies. This has been widely employed for simulations in complex domains. With this approach, the equations become very stiff, and thus the implicit forcing method has been constructed [102]. For the case of interest here, namely flexible interfaces, the explicit forcing method as used in the original immersed-boundary method is sufficient. Even with an explicit time integration scheme, the method is stable given that the time step is sufficiently small [103].

When using the immersed-boundary method to implement a generalised interfacial tension, we follow the procedure given in [104]. The interface is imagined as a continuum of elastic fibres immersed in the fluid. These fibres serve as a device for deriving the model. They do not have a mass nor do they occupy a volume, but together with the fluid they are immersed in, they act as a viscoelastic material. The fibres are arranged in a mesh parametrised by three space coordinates, which we take to be $(\phi, r, s)$ with reference to Figure 3. With this framework, fixing two of the space coordinates, e.g. $(\phi, r)$, uniquely determines a fibre. The last coordinate, $s$, is then a parametrisation along the elastic fibre given by the fixed values of $(\phi, r)$.

For the case of interest here, namely a drop interface that has no thickness, one coordinate is given by the other two, i.e. $r = r(\phi, s)$; for a spherical drop $r$ would be constant. Moreover, we consider the axisymmetric case, meaning that $r = r(s)$ and that nothing depends on $\phi$. This means we consider the situation illustrated in Figure 3, i.e. a single fibre going in the meridional (or $s$) direction of the drop, which is discretised by many points. For each such point there is also a fibre going in the azimuthal direction, which is not discretised.

Using Peskin's notation, we write the strains as e.g. $\lambda_s = |\frac{\partial \mathbf{X}}{\partial s}|$. We then have [104] the forces from the interface acting on the fluid given as

$$\mathbf{f} = \frac{\partial T_s}{\partial s}\mathbf{t} + T_s \left|\frac{\partial \mathbf{X}}{\partial s}\right| \kappa_s \mathbf{n} + T_\phi \left|\frac{\partial \mathbf{X}}{\partial \phi}\right| \kappa_\phi \mathbf{n}. \tag{30}$$

From this we see that the force consists of a component along the fibre in the direction $\mathbf{t}$, as well as a component in the principal normal direction, pointing towards the centre of the osculating circle of the curve, $\mathbf{n}$. As previously noted, there is no force in the binormal direction, $\mathbf{t} \times \mathbf{n}$.

If we assume no elasticity and a constant interfacial tension, $T_s = T_\phi = \gamma$, Equation (30) becomes

$$\mathbf{f} = \gamma \left( \left|\frac{\partial \mathbf{X}}{\partial s}\right| \kappa_s + \left|\frac{\partial \mathbf{X}}{\partial \phi}\right| \kappa_\phi \right) \mathbf{n} \tag{31}$$

and the tangential force disappears. This corresponds to the normal two-phase flow situation with a simple interface described only by interfacial tension, and will serve as a test case for the hybrid method. The deformations entering into this expression serve as normalisation factors, since the original expressions are derived with reference to the undeformed coordinate system. This point may be confusing at first; the reader is referred to the thorough derivation in [104]. The parenthesis in this expression corresponds to the mean curvature of the drop, so the entire expression corresponds to the familiar Laplace-Young formula.

To implement the tensions numerically requires computing the interfacial deformations. Let

$$\|\mathbf{X}\|_i^k = \|\mathbf{X}_k - \mathbf{X}_i\| \tag{32}$$

be the Euclidean distance between Lagrangian points $i$ and $k$. For simplicity we restrict the disposition here to the fully two-dimensional case, where $T_\phi = 0$ and the tension $T$ is given as

$$T_s = K_a \left( \left| \frac{\partial \mathbf{X}}{\partial s} \right| - 1 \right) + \gamma \tag{33}$$

Discretising this equation gives the following expression for the tension at point $i$, $T_{s,i}$:

$$T_{s,i} = K_a \left( \frac{\|\mathbf{X}\|_i^{i+1} + \|\mathbf{X}\|_{i-1}^{i}}{l_i + l_{i-1}} - 1 \right) + \gamma, \tag{34}$$

where $l_i$ is the equilibrium length between point $\mathbf{X}_i$ and $\mathbf{X}_{i+1}$. This gives the tension for each Lagrangian point along the boundary, which is then used in a discretised version of Equation (30),

$$\mathbf{f}_i = \frac{T_{s,i+1} - T_{s,i-1}}{2} \mathbf{t}_i + T_{s,i} \frac{\|\mathbf{X}\|_i^{i+1} + \|\mathbf{X}\|_{i-1}^{i}}{2} \kappa \mathbf{n} \tag{35}$$

to compute the discretised interfacial force. This force enters the right-hand side of the discretised Navier-Stokes equations on the Eulerian grid points close to the interface. The force is smeared out to these points using the mollified delta function, which is made up of combinations of one-dimensional mollified delta functions $\delta(r)$.

In contrast to other smeared-interface methods, the delta function in the immersed-boundary method is uniquely determined by six requirements on the properties of this function. The reader is referred to [72, sec. 6], as well as the previously mentioned MSc thesis by Lysgaard [90], for details of the delta function construction, as well as description of the spreading of Lagrangian quantities to the Eulerian grid, and the interpolation in the opposite direction. The resulting one-dimensional delta function, which is the basis for both the interpolation and spreading operations, is

$$\delta(r) = \begin{cases} 0, & r \leq -2 \\ \frac{1}{8}\left(5 + 2r - \sqrt{-7 - 12r - 4r^2}\right), & -2 \leq r \leq -1 \\ \frac{1}{8}\left(3 + 2r + \sqrt{1 - 4r - 4r^2}\right), & -1 \leq r \leq -0 \\ \frac{1}{8}\left(3 - 2r + \sqrt{1 + 4r - 4r^2}\right), & 0 \leq r \leq 1 \\ \frac{1}{8}\left(5 - 2r - \sqrt{-7 + 12r - 4r^2}\right), & 1 \leq r \leq 2 \\ 0, & 2 \leq r \end{cases} \tag{36}$$

where $r$ is the distance e.g. from a Lagrangian point to the Eulerian grid cell centre.

We remark that if the distance between two Lagrangian points is too big, the spreading operations using the delta function will not approximate the continuous versions correctly. For this reason it is required that two Lagrangian points never be further apart than half the width of an Eulerian grid cell.

In the proposed method, cubic splines are used to generate a smooth analytic parametrisation of the interface. The main advantage of this is that properties such as the curvature, tangent- and normal vectors are all naturally defined for a cubic spline. Figure 4 shows the immersed boundary elements around the point with index $i$ together with their different properties, and where they are defined. The cubic spline fitted to the points is only evaluated at the nodes. Because of this the curvature is only available at these points. The same applies to the normal vectors, which are directly calculated from the first derivative of the spline at the nodes. On the other hand, line segments are computed as the difference in position between two adjacent points. This means that lengths are defined on the segments, and not on the nodes. Note that the cubic spline going through the points is not shown in this figure, but it is used to compute the curvature $\kappa$ and the normal vector $\mathbf{n}$.

It should also be noted that the differences in segment length are exaggerated in Figure 4, since the tangential term in the interfacial force will very quickly eliminate such differences. This is in line with what one expects from such longitudinal waves, which are known from theory [105] and experiments [106] to be

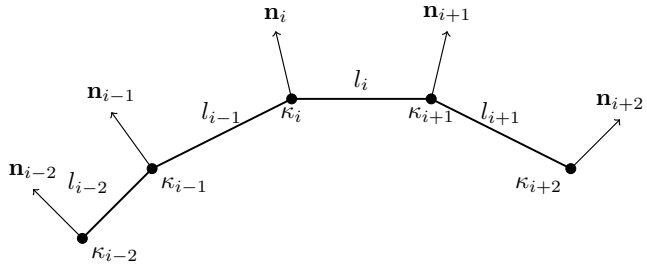



Figure 4: Part of immersed boundary grid showing where different values are located.

extremely rapid. This can also be understood intuitively since, in contrast to a regular capillary wave, a longitudinal surface wave displaces essentially no mass, and thus inertial effects are very small.

When computing the interfacial force on each node, all variables are required at the node. As mentioned, the segment-lengths are not stored at the nodes. An option would be to use the cubic spline to calculate the length, but this requires the numerical evaluation as well as inversion of an elliptic integral. To keep the method simple, we approximate the segment length as the average of the linear distance from the node in question to its two neighbours. The curvature and unit normal vectors remain analytically evaluated from the cubic spline.

In total, this approach encapsulates all the surface effects we need to simulate in one coherent framework. If, say, the elasticity of the material is a function of temperature, or the elasticity is found to be non-Hookean, or some relaxation behaviour is observed that makes the elastic modulus a function of the applied strain, these effects can easily be accounted for by modifying Equation (33).

*2.4.5. Computing the level-set function from the immersed boundary*

When using both the immersed boundary method and the ghost-fluid method to calculate interface forces, special care has to be taken to make the methods consistent. The following technique is proposed for that. The geometry is completely determined by the Lagrangian points along the interface. In each stage of the time integration method, the shortest distance from the Eulerian points to the Lagrangian boundary is computed. In other words, we compute the level-set function purely from the immersed boundary.

This has several advantages. First, advection is moved from the level-set function to the immersed boundary. When no advection of the level-set function is required, it is no longer needed to reinitialise it, Equation (24), or extrapolate the velocity, Equation (23). These routines are somewhat costly, and their saving leads to a $\sim 25\%$ reduction in wall clock run time for some typical two-phase simulations. Second, using this approach, the level-set function is always the best possible approximation to the exact signed distance function for the given Eulerian grid. Third, given equal initial conditions for the immersed boundary and the level-set field, the two descriptions of the interface may not be consistent with respect to each other, meaning that after some time, $t$, the advection of the level-set function and the Lagrangian points could cause the two methods to have slightly different locations for the interface [2]. This is problematic because the interfacial forces would appear at two different interfaces rather than one. This inconsistency disappears when reinitialising the level-set function from the immersed boundary at every timestep.

The algorithm for computing the level-set function from the Lagrangian points is as follows.

↪ Loop over the Lagrangian points representing the interface

  ↪ For each line segment connecting two points, compute its bounding box.

[2]The reason for this is that the immersed boundary points can have sub grid details. This means that inside a grid cell there will be differences between the level set and the immersed boundary. Over time these will grow bigger than one grid cell because of advection. At that point, the two interface descriptions are not consistent with each other.

↪ Grow the bounding box such that it contains the widest Eulerian stencil used in the discretised reinitialisation equation.

↪ Loop over the Eulerian grid points inside the bounding box

  ↪ Compute the shortest distance from this grid point to the line segment using standard formulae.

  ↪ Compute whether the grid point is inside or outside of the closed interface using a standard point-inside-polyhedron algorithm.

  ↪ From these two results, compute the signed distance.

  ↪ If this is the smallest distance computed for this grid point so far in the outer loop, store it as the signed distance for this grid point.

*2.4.6. Penalisation method*

As will be seen in Section 5.2, we want to be able to simulate solid objects in our domain, in addition to the two-phase flow with complex interfaces. To achieve this we utilise a standard $L_2$ penalisation method [107] since it is very easy to implement (literally just twenty lines of code) and since it can also be used to enforce a flow field such as the desired suction inside a pipette. With the penalisation method, the flow field exists inside the solid objects, but is forced to be approximately equal to zero, or in general equal to a specified field $\mathbf{u}_{\text{spec}}$, through a forcing parameter $1/\eta$ which enters in an additional term $(1/\eta)\chi(\mathbf{u}_{\text{spec}} - \mathbf{u})$ added to the right-hand side of the momentum equation. Here the scalar field $\chi$ is a marker function for the solid body, so it is 1 inside the body and 0 outside it. One can think of the penalisation term as an additional body force which outside the body is zero and inside the body is proportional to the difference between the actual and the prescribed flow field. In [107] proofs of the existence and uniqueness of solutions with this method, as well as an error estimate, are given. The error is of the order of $\eta$. To get good results, one would naively set $\eta = 0$ and get zero error, but as the time step needed for stability is proportional to $\eta$ there is the usual trade-off between speed and accuracy.

*2.4.7. The time step restrictions*

In the simulations, the appropriate time step is adjusted dynamically using the conditions given here, in order to have time steps as large as possible without causing the method to become unstable. Following [108, sec 3.8], we take the contribution from the advection term into account with

$$\mathrm{M}_c = \frac{\max u_x + \max u_y}{\Delta}, \tag{37}$$

where $\Delta$ is the width of an Eulerian cell and $\max u_x, \max u_y$ are the largest magnitudes taken by the velocity components in the simulation domain. The contribution of the viscous stress to the time step restriction is taken into account with

$$\mathrm{M}_v = \max\left(\frac{\mu_1}{\rho_1}, \frac{\mu_2}{\rho_2}\right)\frac{4}{\Delta^2}. \tag{38}$$

These are combined with the contribution from the interfacial force $\mathbf{f} = [f_x \ f_y]^T$ to form the time step restriction

$$\frac{\Delta t}{2}\left((\mathrm{M}_c + \mathrm{M}_v) + \sqrt{(\mathrm{M}_c + \mathrm{M}_v)^2 + \frac{4f_x + 4f_y}{\Delta}}\right) \leq C \tag{39}$$

From smearing out the interfacial force density $\mathbf{F}$ using the mollified delta function $\delta_\Delta$, we have that $\mathbf{f} = \mathbf{F}\delta_\Delta$ and since $\delta_\Delta \leq \frac{1}{\Delta}$, cf. Equation (36), the time step restriction can be written as

$$\frac{\Delta t}{2}\left((\mathrm{M}_c + \mathrm{M}_v) + \sqrt{(\mathrm{M}_c + \mathrm{M}_v)^2 + \frac{4F_x + 4F_y}{\Delta^2}}\right) \leq C. \tag{40}$$

In this final condition, $C$ is the time step safety factor, typically $C = 0.5$. Finally, as mentioned in the previous section, when the penalisation method is used, the timestep must also fulfill $\Delta t \leq \eta$ where $\eta$ is the penalisation parameter.

## 3. Summary of the proposed method

At this point we may assemble the proposed multiscale method in its entirety.

- **At the nanoscale:** consider a tiny patch of the interface, $\sim 300$ nm$^2$:
  - A volume around this patch is simulated using coarse-grained molecular dynamics.
  - Accurate models for water, heptane, toluene and asphaltenes are used.
  - The domain is elongated normal to the interface. Large systems of $\sim 10^6$ atoms are simulated.
  - The interfacial tension $\gamma$ is computed from Equation (7).
  - Using volume-preserving deformations, the elasticity $K_a$ is computed from Equation (9).
- **At the macroscale:** two-phase flow simulation of drop with complex interfaces:
  - Flow is governed by Equations (10) and (11), solved using numerical methods described in Section 2.4.
  - The interface is handled with the new hybrid level-set/ghost-fluid/immersed-boundary method.
  - Level-set/ghost-fluid method gives a sharp handling of density and viscosity jumps using Equations (27) to (29).
  - Immersed-boundary method gives accurate interface representation, and computes the tension $T$ with Equations (17) and (18) using $\gamma$ and $K_a$ from the nanoscale.
  - The forces caused by $T$ are computed from Equation (31) and distributed from the Lagrangian points to the Eulerian grid using Equation (36).
  - The level-set function is computed from the Lagrangian points using the algorithm in Section 2.4.5.

In Figure 5 the method is summarised, showing the vectors and interface representation living on the Eulerian grid, the immersed boundary Lagrangian points and the tensions acting on them, and in the corner the molecular dynamics simulation which represents a tiny patch of the interface and is used to estimate the properties $K_a$ and $\gamma$.

We will now proceed to validate the different components that make up this method. We begin in Section 4.1 with the macroscale method, and continue in Section 4.2 with the nanoscale, where the models for different fluids and for the asphaltene molecules are considered.

## 4. Validation

We will now present validation results for both the nanoscale and the macroscale models. We first consider the hybrid macroscale method and demonstrate that this method gives the correct results on several test cases. Then we consider the nanoscale models for both the simple fluids and the more complex asphaltene molecules.

### *4.1. Macroscale: validation*

In this section we will demonstrate the validity of the developed hybrid method. We start by demonstrating the superior resolution of the immersed-boundary method over the standard level-set method, as previously mentioned. To this end we employ two standard test cases for interfacial advection, namely the drop in vortex test and Zalesak's disk test.

Following this, we consider the case of an initially elongated spheroidal drop relaxing under interfacial tension. The case is considered both for a two-dimensional and for an axisymmetric drop, and it is also considered both with and without density and viscosity differences across the interface. In summary, we demonstrate that the methods converge to the same solution under grid refinement. For further verification, see [90]. For the continuum simulations presented in this work, we provide tables with details of the configuration and parameters used in Appendix A.

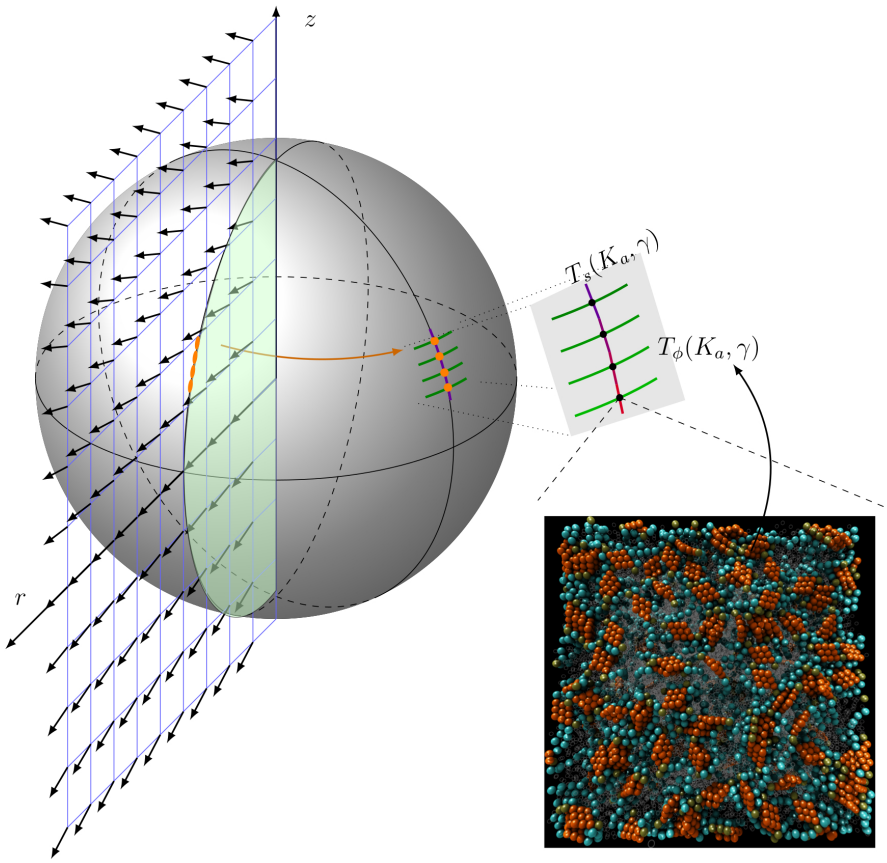



Figure 5: Illustration of the proposed method. On the Eulerian $(r, z)$ grid, the flow field $\mathbf{u}$ (vectors) and the level-set function $\phi$ representing the green-shaded portion of the droplet are shown. Some of the Lagrangian immersed-boundary points are shown in orange, with lines as before indicating the tensions $T_s$ and $T_\phi$. For a tiny patch on the interface, we compute the interfacial properties $K_a$ and $\gamma$ using molecular dynamics simulations (lower right corner). These properties are used in the calculations of the tensions. Note that the colours used in the molecular dynamics simulation snapshot do not have any relation to the other colours in the figure.

*4.1.1. Drop in vortex*

A standard test of advection for interface-tracking methods is the drop in a potential vortex [109]. Here a drop is placed in the unit box, and a static potential vortex advects it. The velocity field is given by

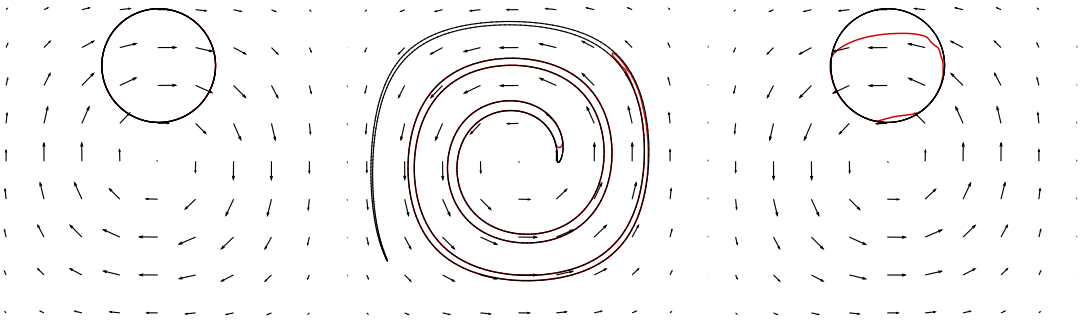
$$u_x = -2\big[\sin(\pi x - \pi/2)\big]^2 \cos(\pi y - \pi/2) \sin(\pi y - \pi/2) \tag{41}$$

$$u_y = -2\big[\cos(\pi y - \pi/2)\big]^2 \sin(\pi x - \pi/2) \cos(\pi x - \pi/2) \tag{42}$$

The remaining parameters for this test are given in Table A.1.

At some time $t = T/2$, the flow field is reversed, and the simulation is run until $t = T$. Then the initial interface is compared with the final one. Figure 6 shows the initial condition (a), the interface at half time (b) where $t = T/2 = 3.5$, and the final interface for both the level-set and the immersed boundary method, in red and black colours, respectively.

We see that the immersed boundary method has no visible mass loss, while the level-set representation loses mass when the drop gets stretched thinner than a grid cell. The reason for the large level-set mass loss



(a) $t = 0$ (b) $t = 3.5$ (c) $t = 7$

Figure 6: Drop in potential vortex. Red is level-set solution while black is the immersed boundary. Velocity vectors are shown for every 20 grid points in each direction.

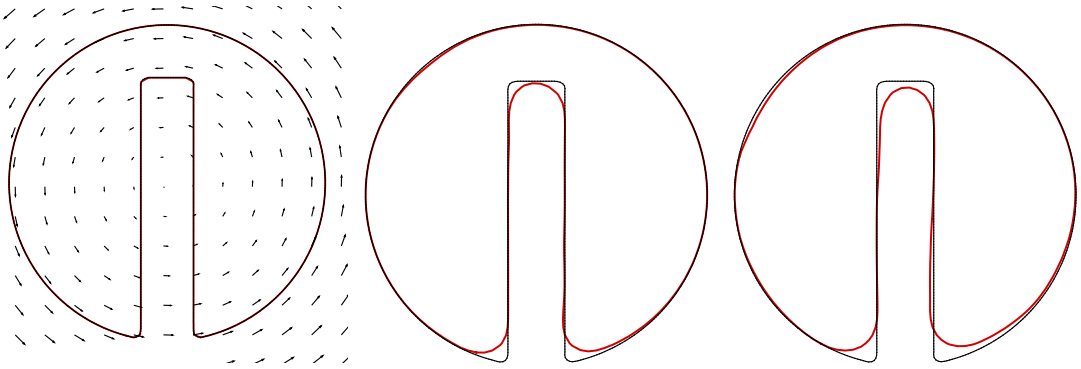
is that when two interfaces come this close together, the discrete level-set function does not have the required resolution to switch sign. This follows from the Nyquist-Shannon theorem. The immersed boundary method does not have this restriction. If one wanted to represent the smaller features with level-set representation, one choice would be to double the grid resolution. For two dimensions this would make the computational cost increase quadratically. To get the same increase in resolution with the immersed boundary method, one would need to double the number of points, this would only double the amount of work needed. Thus immersed boundary scales considerably better than level-set with respect to the interface resolution.

The previous argument makes immersed boundary seem superior to level-set when it comes to resolution. However, this is not the whole story. For the immersed boundary to represent a non-smooth sub-grid feature, the Lagrangian points have to be advected in a sub-grid way. With the immersed boundary method, the Lagrangian points are advected using an interpolated of the velocity field from the Eulerian grid. This means that the highest wave number that can be created in the immersed boundary representation *by the flow* is proportional to $1/\Delta$ (where $\Delta$ is the width of the Eulerian grid cells). For smooth velocity fields, like this potential vortex, the immersed boundary method has some sub grid resolution. This is because it can accurately represent stretching, squishing and other smooth transformations that lead to sub grid details. Also, in the case of the interface crumpling as will be discussed in later sections, the immersed boundary representation will produce wrinkles inside each grid cell.

*4.1.2. Zalesak's disk*

Another interesting difference between an Eulerian and Lagrangian description of geometry is the effect of grid alignment. This effect can be seen in the next test, Zalesak's disk [110]. Here, a slotted disk is put in a velocity field that has constant angular velocity, corresponding to rigid body rotation. The boundary is advected one or several revolutions and the result is inspected. Further details of the case are given in Table A.2.

From Figure 7 it is clear that the immersed boundary resolves the rotated disk better than the level-set function. During the rotation, information is lost in the level set, while the immersed boundary is virtually not affected. The reason for this is that the level set, based on an Eulerian grid, cannot represent non-smooth features that are not aligned with the grid perfectly. This means that over the duration of the rotation, small errors in the interface position creeps in as a consequence of the interface not being straight and aligned with the grid. In the immersed boundary method, the grid has no preference about the orientation of the interface. When it comes to the drop-in-vortex test, Section 4.1.1, one may argue that the difference between the two methods is exaggerated by the specifics of the test, to the detriment of the level-set method. There does not seem to be any such argument for Zalesak's disk. The immersed boundary method is fundamentally better at preserving non-smooth features like corners without smearing. In real life, non smooth interfaces occur e.g. when two drops coalesce.



(a) $t = 0$ (b) $t = 40$ (c) $t = 80$

Figure 7: Zalesak's disk for 0, 1 and 2 revolutions. Red shows level-set interface while black shows immersed boundary. The velocity field is constant in time and corresponds to rigid body rotation.

*4.1.3. Comparison with reference method*

We have now verified that the immersed boundary method captures the interface correctly under advection. Next we need to verify that the forces from the boundary on the fluid are implemented correctly. The handling of the viscosity and density jumps must also be verified to be correctly coupled with the immersed boundary. The technique chosen for this was to compare the proposed method, Section 3, with a reference method, the level-set method with the ghost fluid method, which has previously been verified and validated [84, 85, 86, 87, 88, 89]. To have some measure of the drop dimensions during oscillations, the horizontal and vertical axis lengths are used. These parts of the drop are the ones most rapidly advected, with the highest pressure differences and curvatures. Thus any differences between the two methods would be most pronounced at these points.

In the first test, an ellipsoidal drop is relaxing to its equilibrium, a sphere, driven by interfacial tension. The purpose of this test is to verify that interfacial tension simulated with the proposed hybrid method gives the same results as when simulated with the standard level-set/ghost-fluid method. There is no gravity and no density or viscosity differences in this test. This way, all forces are generated by interfacial tension as the ellipse relaxes to equilibrium. The parameters of the test are listed in Table A.3. The test was run for increasing grid resolutions to see how the two methods compare under refinement. The result for the axisymmetric simulation can be seen in Figure 8. We see that under grid refinement, the proposed method converges to the same answer as the previously verified and validated implementation of the level-set/ghost-fluid method. This demonstrates that the hybrid level-set/ghost-fluid/immersed-boundary method gives the correct result. For corresponding tests in two dimensions, see [90].

This test shows that the proposed method converges to the same solution as the reference method for a relaxing ellipse driven by interfacial tension. For the coarsest grid, it appears the hybrid method is the least accurate. This is likely caused by the disagreement between the Eulerian and Lagrangian interface representations becoming significant when the curvature of the interface is no longer much smaller than $1/\Delta$.

This test confirms that the method is consistent, but there is no jump in density or viscosity in this case. As discussed previously the proposed method will treat density and viscosity jumps in a sharp fashion, while the tension in the interface is handled in a smeared-out fashion. To verify that this also gives consistent results, a simulation of a similar case, i.e. a relaxing ellipse driven by interfacial tension was set up. Instead of equal density and viscosity, the density ratio is now 2 and the viscosity ratio is 10. These parameters are representative for the case of a water drop in oil. The simulation was run on a moderately fine grid, $N = 400$, where good agreement was found between the two methods in the previous test. The full set of parameters for the simulation are listed in Table A.4. The simulations were run both for 2D and axisymmetric flow for both methods.

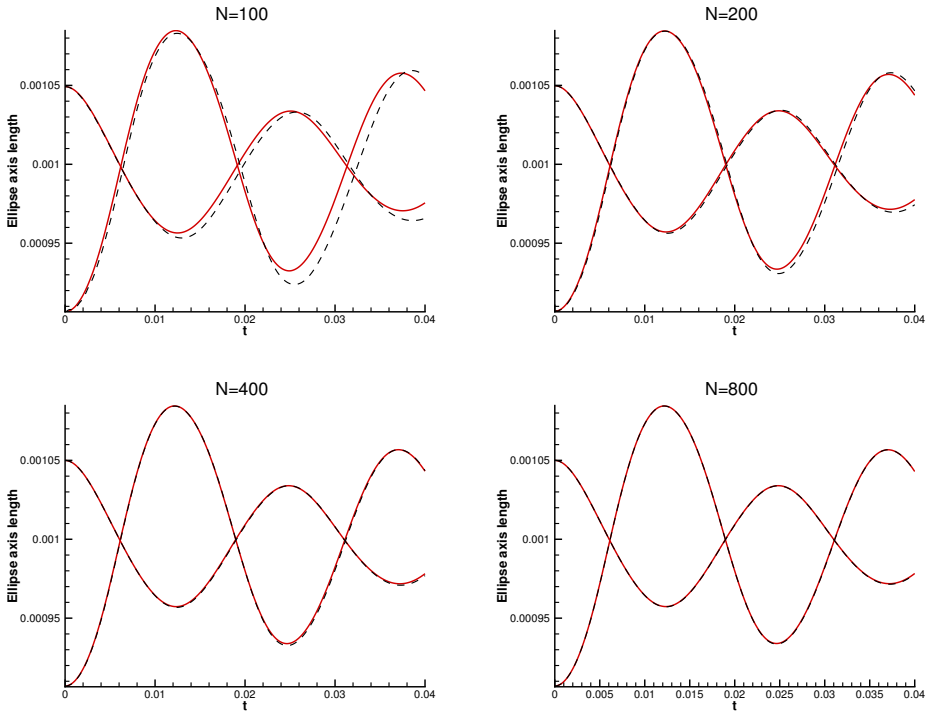
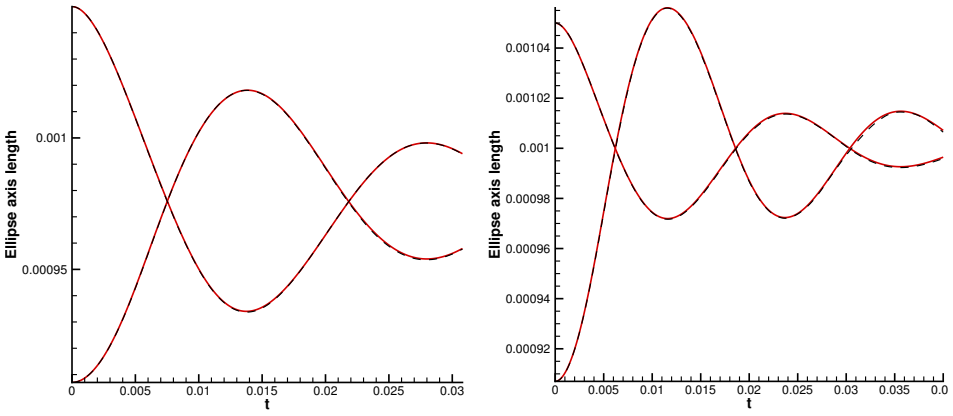



Figure 8: Drop axis lengths for the axisymmetric relaxing drop. Red is the reference solution, dashed black is immersed boundary solution. The two methods converge to the same solution as the grid is refined.




(a) Two-dimensional drop.

(b) Three-dimensional, axisymmetric drop.

Figure 9: Comparison of drop axis lengths for the reference method and proposed method with a viscosity and density jump. Red is reference method while dashed black is proposed method.

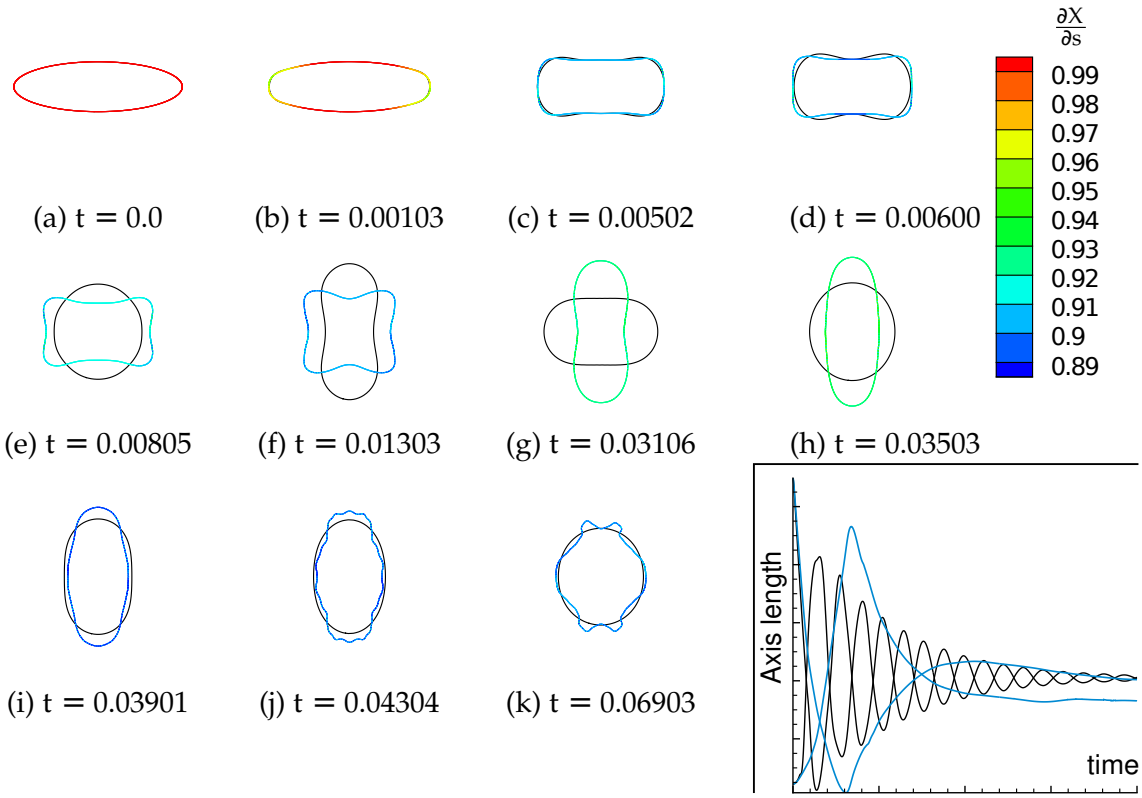



Figure 10: Several frames of the two-dimensional simulation with elastic membrane, (coloured) together with the clean interface, (black). The colours indicate the relative length of the interface compared to its equilibrium length. The inset in the lower right corner shows the time evolution of the horizontal and vertical axis of the clean (black) and the contaminated (blue) interface, which indicates that clean drop undergoes many oscillations during this time, while the contaminated drop only undergoes one.

As seen in Figure 9 the two methods are in agreement both for two-dimensional and axisymmetric flow. This shows that the proposed method correctly and consistently combines the interfacial tension from the diffuse interface, with the sharp handling of viscosity and density jumps.

*4.1.4. Relaxing drop with elastic membrane*

To test the effect of additional elasticity on the interface, the relaxing ellipse was again considered. In this two-dimensional test case, the initial condition is an ellipse with both interfacial elasticity and interfacial tension, compared to the same case with zero elasticity. The parameters for this test are listed in Table A.5. At the initial state, the deformation $\frac{\partial \mathbf{X}}{\partial s} = 1$ in Equation (33), so the elasticity does not contribute to the interfacial force. One can say that from the elastic point of view, the interface is neither stretched nor compressed, but there remains the constant interfacial tension which produces a force. In this particular case, $K_a$ is set to be ten times larger than than $\gamma$, which means that $T$ computed from Equation (33) will be close to zero when $\frac{\partial \mathbf{X}}{\partial s} \approx 0.9$. In other words, when the membrane is compressed to 90% of its original length, elastic forces and interfacial tension forces will be in balance.

The interface starts in the initial state shown in red in Figure 10 (a). Because of its eccentricity, interfacial tension is relatively strong on the left and right sides of the drop and it is quickly compressed, Figure 10 (b). After $5 \times 10^{-3}$ s the drop is compressed to approximately 90% of the initial length Figure 10 (c). This means that the interface no longer introduces any force, and without any viscosity or density differences the simulation would proceed as if it was momentarily single phase. As there is a significant flow still present

in Figure 10 (c), advection of the interface continues. Some of the kinetic energy which is not dissipated by viscosity goes into deforming and again stretching the interface. At $t = 3.5 \times 10^{-2}$ s, Figure 10 (h), the interface is stretched to the next maximum again, and the velocities are close to zero. Now there is not enough potential energy in the membrane to do another oscillation, it is in some sense overdamped. The interface contracts creating a crumpled drop as seen in Figure 10 (i) to Figure 10 (k). As this has happened the drop with a clean interface, in black, has oscillated towards its equilibrium shape, a circle, by going through about eight oscillation cycles.

It is clear that for these parameters, the elastic membrane has a significant effect on the time evolution of the drop, fundamentally changing its response. For clean fluids, the equilibrium interface is always the one that has the minimal interface area. The interfacial elasticity changes this situation, and the equilibrium state is no longer obvious given the initial conditions. One insight from this simulation is that for a clean drop without an elastic membrane, there exists a unique spherical equilibrium state, only given by the initial volume of the drop. On the other hand, for the drop with an elastic membrane, the equilibrium is not just a function of the initial volume, but also of the initial shape. This is because the initial shape affects what parts of the drop are stretched and compressed, which has a significant impact on the final steady state. This demonstrates that the evolution of a drop with an elastic membrane is more complex than one without.

### *4.2. Nanoscale: validation*

When running simulations of a complex multi-component system, the first thing to consider is each of the two-component systems. For the toluene-water and heptane-water systems, there is one free parameter, the binary interaction parameter $k_{ij}$, to tune in each case. There is also an important value to tune this against, namely the interfacial tension measured in experiments. For the alkane-water system with the models used here, Lobanova et al. [111] obtained the value of $k_{ij} = 0.3205$, transferable across the alkanes with different lengths. The water [44] and alkane [111] models used have been published previously. The model for toluene used here has not been published previously; see Appendix B for force field parameters for this model.

For the toluene-water system, such tuning has not been done previously, so it is done here. The system consisted of 10 000 toluene molecules and 40 000 water molecules, placed in an elongated box, and simulated at 20°C and 1 bar. After obtaining the desired temperature and pressure by simulating in the $NpT$ ensemble, the system was allowed to phase-separate such that two slabs of liquid were formed. Subsequently, the system was simulated in the $NVT$ ensemble for 50 nanoseconds to obtain the interfacial tension.

The three values for the cross-interaction parameter, $k_{ij} = (0.1, 0.2, 0.3)$, were initially tested, and subsequent guesses were refined until the value $k_{ij} = 0.241$ was found, which gave an interfacial tension in very good agreement with the experimental value of 37 mN/m [112] at this temperature and pressure. See Figure 11, where the cumulative average of the interfacial tension computed from Equation (7) is plotted as a function of the simulation time. Also plotted is the experimental result (dashed orange line) and the running average over 1000 values of the interfacial tension (blue dots).

Having the cross-interactions for the two binary systems established, verifying that the interfacial tension for the heptane-toluene or "heptol" mixture against water agrees with experiments is a good demonstration of the predictive power of the method. When discussing a heptol mixture, one must distinguish between molar ratios, convenient in simulations, and volume ratios, convenient in experiments. To avoid confusion, we will refer here to volume ratios using the notation $N : M$, and molar ratios using the notation $n/m$. The molar mass of heptane and toluene are 100.2 and 92.1 g/mol, respectively, and their densities are 684 kg/m$^3$ and 867 kg/m$^3$, respectively. This means that a 50/50 molar ratio gives a volume ratio of 1.38:1, and conversely, a 1:1 volume ratio gives a 42/58 molar ratio.

To test the prediction of interfacial tension of the heptol-water system, a corresponding simulation was set up with a 1:1 heptol mixture against water at 20°C and 1 bar. Experimental data for this interfacial tension is not available, but an accurate estimate of it can be obtained following Yarranton and Masliyah

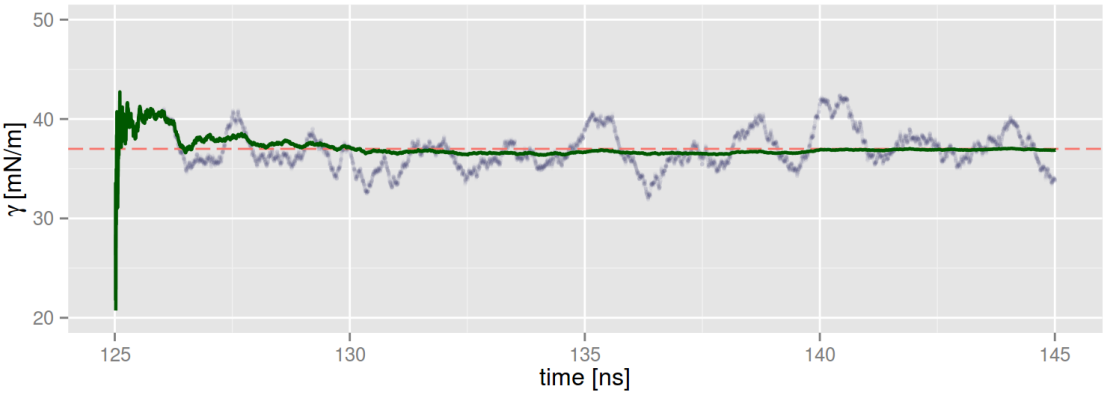



Figure 11: Interfacial tension of toluene and water, using the cross-interaction parameter $k_{ij} = 0.241$. The green line shows the cumulative average, the blue dots show the running average over 1000 points, and the dashed orange line shows the experimental value.

[113]. This procedure gives the interfacial tension of an organic mixture against water as

$$\gamma_{\text{mix}} = \gamma_2 - R_u\, T\, \Gamma_m \ln\left(1 + x_1(\exp\left(\frac{\gamma_2 - \gamma_1}{R\, T \Gamma_m}\right) - 1)\right) \tag{43}$$

where $\gamma_1$ and $\gamma_2$ are the two pure-component interfacial tensions against water, $R_u$ is the universal gas constant, $T$ is the temperature in Kelvin, $x_1$ is the molar fraction of component 1 in the mixture. We denote here toluene as component 1 and heptane as component 2. The interfacial tensions are $\gamma_1 = 37$ mN/m [112] and $\gamma_2 = 51.2$ mN/m [114]. The parameter $\Gamma_m$ is estimated as $\Gamma_m = 0.00415$ mmol/m$^2$ by Yarranton and Masliyah [113], which is shown to give good results for a wide range of alkane-aromatic mixtures. With these expressions, the interfacial tension of the mixture is computed as $\gamma_{\text{mix}} = 41.9$ mN/m.

The simulation result is plotted in Figure 12, together with $\gamma_1$, $\gamma_2$ and $\gamma_{\text{mix}}$. The simulation result matches very well the value computed from Equation (43). Note that the cross-interaction between heptane and toluene has not been tuned at all. This illustrates the predictive power of the SAFT-$\gamma$ Mie approach, i.e. that one can compute, with good accuracy, physical properties that have not been used when constructing the model.

With the simple fluids taken into account, we may proceed to consider the models for the asphaltenes. As discussed in the introduction, asphaltenes are immensely complex mixtures of different molecules that together form a solubility class. It is likely that an asphaltene molecule in solution never interacts with an identical molecule, and thus modelling the asphaltenes as a single molecule is a significant simplification. It is likely that improved asphaltene models must take the polydispersity explicitly into account, i.e. having many different types of model asphaltenes in the same simulation. This may be considered in future work.

Molecular dynamics simulations of asphaltenes in the bulk have been considered in several previous works, see e.g. refs. [115, 116, 117, 118, 119, 120, 121, 122, 123]. In these studies, the association and aggregation of asphaltenes have received particular focus, as this behaviour is very important for asphaltene deposition in rocks and in pipelines. Of particular note is the work by Boek et al. [120], where the technique of quantitative molecular representation (QMR) is used to compile a set of model asphaltene molecules, based on several experimental sources of information such as mass spectrometry, NMR, X-ray and neutron scattering studies. These model asphaltenes are employed by several subsequent authors, and have influenced the construction of the coarse-grained model asphaltenes used in this work. Sedghi et al. [122] studied the effect of different side groups and substitutions on one of these representative asphaltene molecules, and found that small changes to the chemical composition caused large variations in the aggregation behaviour.

Only very recently have molecular dynamics simulations been applied to the study of asphaltenes at the

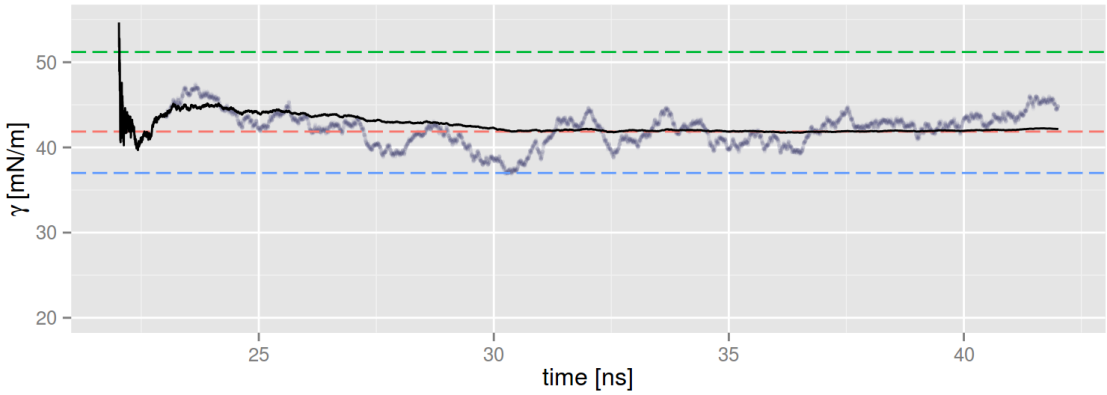



Figure 12: The interfacial tension for a 1:1 heptol mixture against water, computed from molecular dynamics simulation. Black line: cumulative average of the interfacial tension. Blue dots: running average of the interfacial tension over 1000 samples. Orange dashed line: $\gamma_{\text{mix}}$ predicted from Equation (43). Blue dashed line: interfacial tension of toluene-water. Green dashed line: interfacial tension of heptane-water.

oil/water interface. Mikami et al. [124], Liu et al. [125], as well as Yang et al. [126] used atomistically-detailed simulations to study this system. In [124, 125], the previously mentioned QMR-based model asphaltenes were employed, whereas in [126], two different model asphaltenes were proposed based on experimental measurements and used in simulations. These studies highlight the significant challenge encountered when using an atomistically-detailed approach: one is either confined to very few asphaltene molecules [126] or very short timescales [124, 125] due to the high computational cost. A coarse-grained approach, such as that used here, can provide a solution to this; see e.g. the review by Brini et al. [38] for an overview of the advantages of coarse-grained simulation for soft matter systems in general. Recent work by Ruiz-Morales and Mullins [127] employed a coarse-grained (dissipative particle dynamics) approach, but this study is also limited to considering few molecules and short time scales. Finally, it should be noted that in all these studies, the asphaltenes are initially placed at the interface.

In contrast to these limitations of either few molecules or short time scales, and enforced interfacial adsorption, we consider here simulations with up to two orders of magnitude more asphaltene molecules than considered in [126] in conjunction with two orders of magnitude longer simulation times than considered in [125]. Also the asphaltenes are not initially placed at the interface, but randomly distributed throughout the oil phase. Thus the interfacial activity is an inherent property of the model asphaltenes used here.

With this in mind, we briefly discuss the three different asphaltene models that were used in the present work. All these models are of the continental asphaltene type, i.e. they have a central core made up of aromatic rings, to which aliphatic tails are attached. The first model asphaltene, presented by Müller et al. [49], was used in the initial studies; we will refer to this as the APCH asphaltene. It is based on using anthracene beads for the aromatic core, which also contains a pyridine ring, and three tails each made from dodecane. The model behaves reasonably like an asphaltene. Results obtained with this model are discussed in Section 5.1.1.

If anything, the APCH model appears slightly too self-associative. This, together with the some advances in the SAFT-$\gamma$ Mie theory for aromatic compounds, led to the second flavor, developed partly in this work. The main difference with the first flavor is that the aromatic beads, which make up most of the core, are obtained using a version of the SAFT-$\gamma$ Mie approach that is tailored for ring-shaped molecules, as opposed to the standard version which assumes they are linearly shaped. The structure is otherwise very similar to the APCH model asphaltene. The parameters for the aromatic core beads are given in Appendix B.

Two different architectures were considered for this new flavor of asphaltenes, built from the same coarse-grained beads, but with different shapes for the aromatic core. Both have three aliphatic tails made from

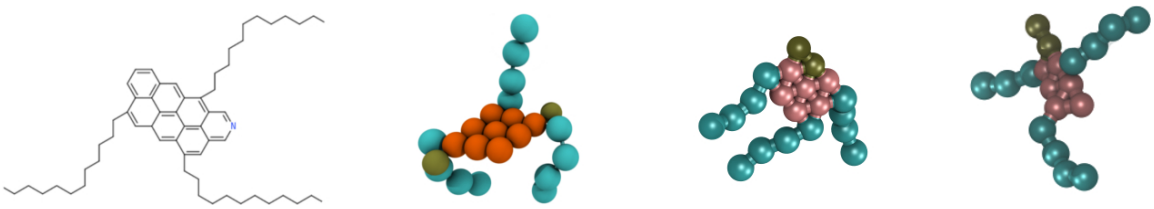


Figure 13: From left to right: corresponding chemical structure, the APCH model, the APCE model and the APCL model. Blue/green beads indicate the aliphatic tails, orange/red beads indicate the aromatic rings forming most of the core, and the tan/gold beads indicate the pyridine.

dodecane, and cores made from six beads representing six hexagonal aromatic rings, together with two beads representing a pyridine, i.e. a hexagonal aromatic ring with a single nitrogen substitution. The difference lies in that "APCE" has a more circular core, while "APCL" has a more elongated core. See Figure 13 for a comparison of the three different model asphaltenes considered in this work.

As mentioned in the introduction, the only usable definition of asphaltenes is that they are insoluble in heptane and soluble in toluene. Accordingly, it should be verified that the new model asphaltene molecules behave in this way. To test this, simulations were run with 240 asphaltene molecules in a system with 40 000 solvent molecules (heptane or toluene) at 20°C and 1 bar.

These simulations gave some interesting results which demonstrate how delicate the energy balance is in systems containing asphaltenes. The two models APCE and APCL appear very similar; however, their solubilities in heptane are completely different, as illustrated in Figure 14 which shows the behaviour in heptane. In this figure, the solvent and the aliphatic tails are omitted for clarity, so one can easily see the stacking of aromatic rings where the molecules cluster. It is readily apparent that the circular core asphaltenes cluster in heptane, while elongated core ones are soluble in heptane. It is also seen that the circular-core asphaltenes cluster with a distribution of a few large clusters, many small clusters, and some monomers; in particular, the asphaltenes do not all gather in one big cluster. This is in agreement with results from experiments (e.g. [128, 129, 130]) and atomistic simulations [119]; and it is in general agreement with the accepted Yen-Mullins model of asphaltene behaviour [131].

The APCE model thus passes the first hurdle, being insoluble in heptane. Simulations of this model in toluene also showed the correct behaviour, as illustrated in Figure 15 where the system is compared in heptane (red) and in toluene (blue) after 50 ns of equilibration. In toluene, some pairs of asphaltenes occasionally come in contact, but inspection of the trajectories showed that these pairs only stay together for about one nanosecond before they break apart again, and thus no larger clusters have time to form.

To summarise, the APCE model asphaltene fits the definition of an asphaltene molecule, forming clusters in heptane but staying in solution in toluene. In heptane, clusters of up to four molecules form after 50 ns. It is likely that five- or six-molecule clusters may form after even longer times or at higher concentrations. A closeup of a four-molecule cluster is shown in Figure 16.

## 5. Results

### *5.1. Molecular dynamics simulations of asphaltenes in oil/water systems*

The crude oil/water system has a very large (and not fully understood) parameter space, so we restrict our attention here to only a few simplified cases. The parameters that can be varied include temperature, pressure, the mixture of alkanes and of aromatics for the fluid components, the ratio between alkanes and aromatics in the fluid components, the amount of resins, the amount of asphaltenes, etc. The liquid components of a crude oil contain a range of alkanes that may extend all the way from methane (liquid at high pressure) to alkanes with 20-30 carbon atoms (liquid at high temperature). Effective models for these mixtures can be obtained by taking a true boiling point curve and dividing it into classes of pseudo-components, e.g. a mixture of $C_5H_{12}$, $C_{10}H_{22}$, $C_{15}H_{32}$, $C_{20}H_{42}$. The very simplest version of this, which we adopt here, is to use just one alkane, for which heptane is a common choice. Similarly, for the aromatic liquids there is a

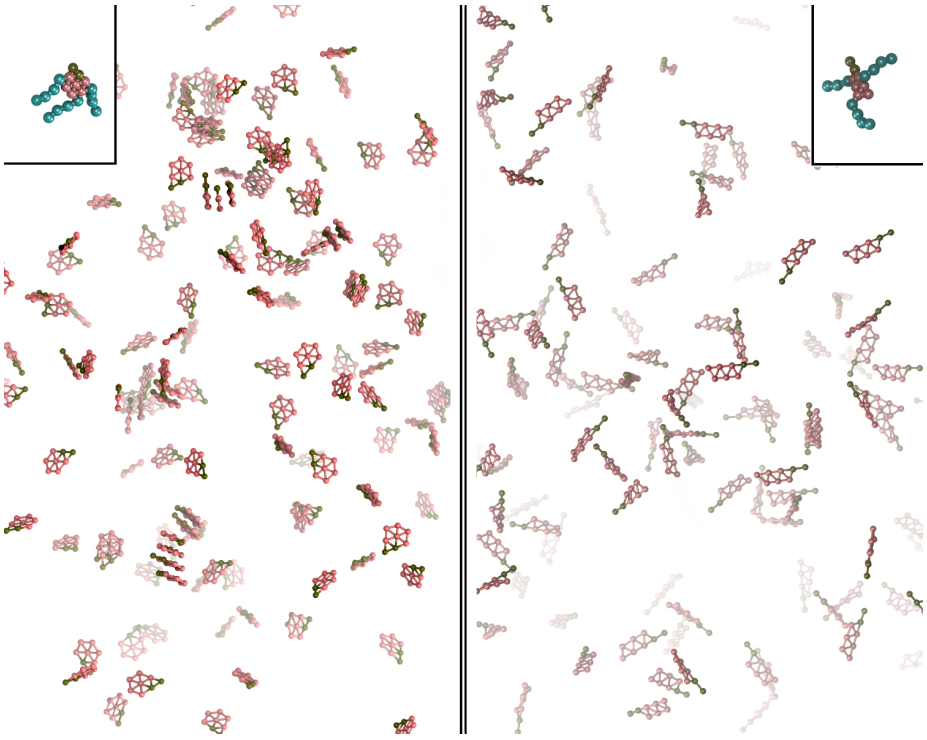


Figure 14: Comparison of the model asphaltenes in heptane. Left: APCE with a circular core. Right: APCL with an elongated core. In both cases, the inset shows the usual coarse grained beads for the entire molecule, while the main figures shows only the aromatic beads and their bonds, with the diameter set to $0.5\sigma$ to make the visualisation clearer.

range to choose from, e.g. benzene, pyridine, toluene etc., and again we limit ourselves to just toluene. This combination, heptol, has been widely used in experimental studies as a model system in which asphaltenes can be dissolved (e.g. refs. [132, 133, 134, 135, 136, 137, 138]). Heptol is the simplest solvent for asphaltenes where the aromatic to aliphatic ratio of the solvent can still be varied.

Given heptol as a base, the mixture ratio can be varied, and asphaltenes can be added to the system in varying concentration. Previous work [132] has indicated that a heptol ratio around 70/30 gives the most stabilising (and thus most interesting) interfacial properties, so the 70/30 heptol mixture is used as a base fluid here. The temperature and pressure are set to 20°C and 1 bar, respectively. The asphaltene concentration is varied, to study the effect this has on the system.

It is notable that in all the simulations reported here, the systems are started from random initial conditions. In particular, the asphaltenes are not placed at the oil/water interface, but they are themselves interfacially active. The tuning of cross-interactions is as reported in Section 4.2 for the heptane-water and the toluene-water cross interactions. For the asphaltenes, the alkane tail-water cross interactions use the same $k_{ij}$ as for heptane-water, otherwise the cross-interactions have not been adjusted.

These studies using the various model asphaltenes were performed at different concentrations from 240 to 960 asphaltene molecules in 40 000 molecules of solvent (70/30 heptol), together with 160 000 molecules of water. These systems were simulated in the $NpT$ ensemble using an elongated simulation box with a 3:1:1 side ratio, and the desired state (20°C, 1 bar, oil/water phase-separated) was obtained. Subsequently, the system was evolved in the $NVT$ ensemble in order to let it equilibrate.

At this point, some remarks are in order with regards to the relaxation times in these systems, as compared to experiments. Looking at the time it takes to reach equilibrium for the interfacial tension in asphaltene-heptol-water systems in experiments (e.g. [136, 139]) the timescales are of the order of hours,

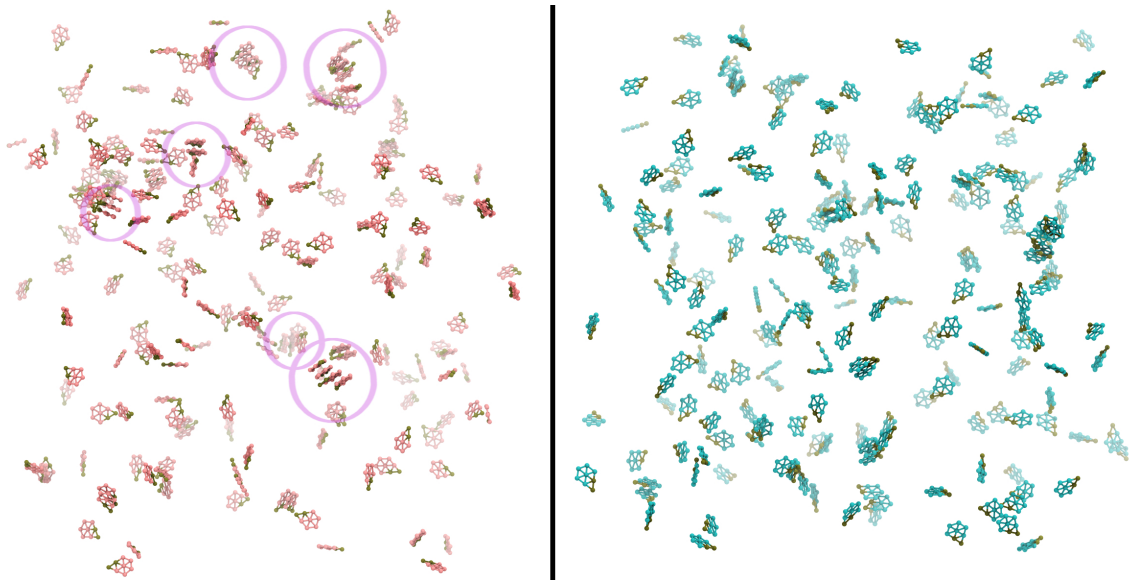


Figure 15: Comparison of the APCE model asphaltenes in heptane (left, red) and in toluene (blue, right). As in the previous figure, only the aromatic cores are shown. Purple circles highlight clusters with three or more molecules (there are none on the right-hand side). It is evident that the molecule is poorly soluble in heptane, where two-, three- and four-molecule clusters can be seen. In toluene, there is no clustering apart from the occasional contact between two asphaltene cores.

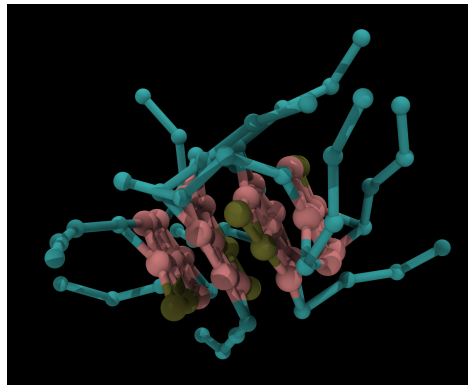


Figure 16: A closeup of a four-asphaltene APCE cluster in heptane. The beads are also here shown at reduced size for clarity; aromatic beads with diameter set to $0.5\sigma$ and alkane beads with diameter set to $0.25\sigma$. The colours indicate aromatic beads (red), pyridine beads (gold) and aliphatic beads (blue)

which is completely out of reach for molecular simulation. However, these long timescales are caused by the slow diffusion of asphaltene molecules from the far-away regions of the bulk to the asphaltene-depleted oil layer close to the interface. Moreover, these timescales are not representative of the situation when water drops travel through oil in a pipeline or a separator, where the ratio of the diffusion boundary layer width to the viscous boundary layer width, i.e. the Schmidt number, can be of the order of $10^6$, meaning that the speed of diffusion is increased by this order of magnitude for a falling drop as compared to a drop at rest in a tensiometer. Comparing this to the situation in simulations, it is clear that the diffusion cannot happen over a longer scale than the simulation box size, which is of the order of 10 nanometres. Experimental measurements indicate the diffusion coefficient of asphaltenes in solvents like heptol is of the order of $10^{-10}$ $\mathrm{m}^2/\mathrm{s}$ [140].

This means the characteristic time for an asphaltene to diffuse across the simulation box is of the order of $10^{-6}$ seconds.

There are two other relaxation time scales of importance, in addition to the time scale for diffusion. The second is the time scale for the adsorption of an asphaltene at the interface. With a coarse-grained description such as here, this time scale is very short, of the order of 1 nanosecond. The third is the time scale for reorientation of the asphaltenes at the interface, i.e. that those molecules already adsorbed find a tighter packing, which allows further molecules to adsorb at the interface. This time scale depends on how the asphaltenes associate at the interface, and is thus difficult to estimate *a priori*.

*5.1.1. Studies using the APCH model asphaltene*

The APCH model asphaltene was used in the first studies of asphaltene-heptol-water systems. Two concentrations were considered, namely 240 and 720 asphaltene molecules in 40 000 molecules of 70/30 heptol. The former corresponds to an asphaltene concentration of about 5%. Note that in these simulations, a simpler two-bead model for the toluene molecule [45] was used.

Snapshots of the oil-water interface from simulations at this lower concentration are shown in Figure 17. This figure shows the interface from the water side after 35 ns (left) and 350 ns (right). The water beads are not shown, and the solvent beads are shown only as grey outlines. It is seen from this figure that the interfacial configuration of asphaltenes changes during this time, with clusters forming at the interface. The number of asphaltenes at the interface increases by about 30% from 35 to 350 ns, and at 350 ns there are 0.25 molecules per square nanometre. Inspection of this system revealed that the interfacial adsorption left the bulk oil almost depleted of asphaltenes. Thus this situation with a lower concentration of asphaltenes is representative of a freshly-formed interface where diffusion has not yet had time to bring in more asphaltenes from the bulk.

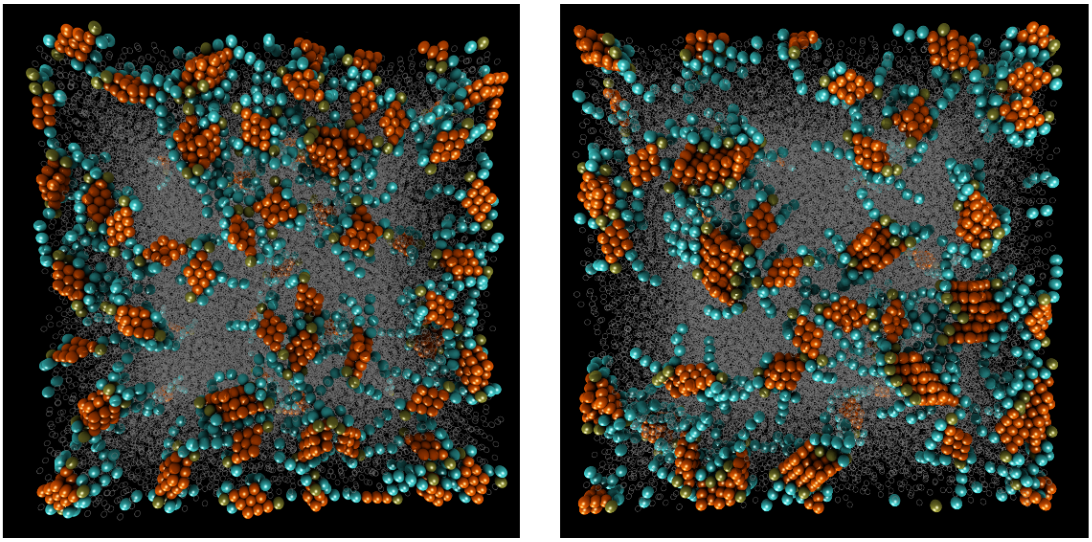


Figure 17: Snapshot of the interface seen from the water side, for the system with 240 APCH model asphaltenes after 35 ns (left) and 350 ns (right). Colours as in previous figures. Water molecules are omitted for clarity, and the heptane and toluene beads are shown in gray.

As previously mentioned, the time scales for diffusion can be of the order of minutes and hours, which is not tractable in these simulations. However, the effect of diffusion may be taken into account by starting the simulation with an initially higher asphaltene concentration. The higher concentration case, using 720 asphaltene molecules, gave about 5% asphaltenes remaining in the bulk after 350 ns, and is thus representative of the same system having reached equilibrium after diffusion from the bulk. Snapshots of this system are

shown in Figure 18, again with 35 ns at the left and 350 ns at the right. Again, there is an increase of about 30% in the number of adsorbed asphaltenes from 35 to 350 ns, and at 350 ns there are 0.5 molecules per square nanometre. It is evident from the snapshot that these asphaltene molecules are highly self-associative at the interface, i.e. that they cluster together. It is likely that this behaviour is not representative of real asphaltenes, which are believed to adsorb with the aromatic core towards the water [141].

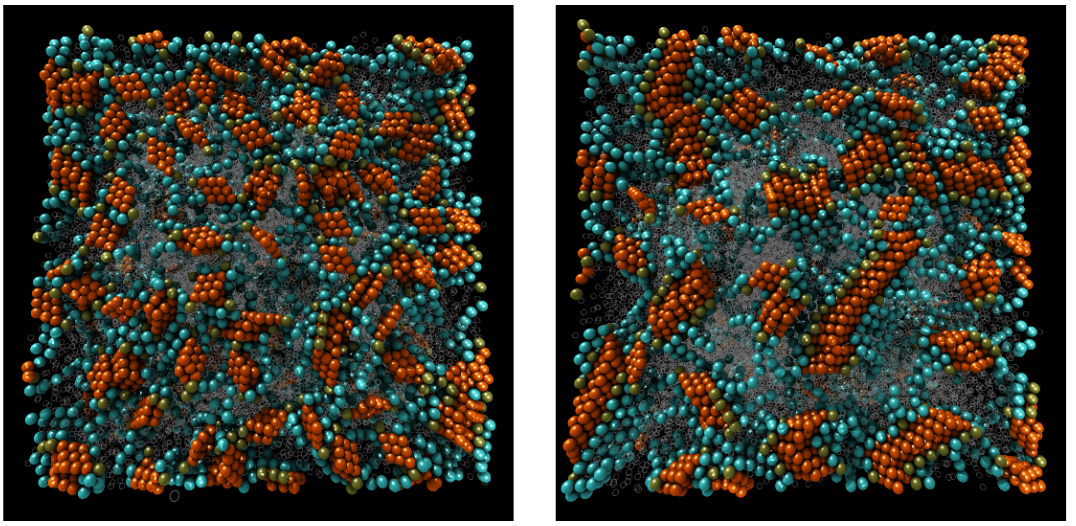


Figure 18: Same as the previous figure, for the system with 720 APCH asphaltene molecules. At this higher concentration, the tendency for the asphaltenes to cluster together at the interface is even more pronounced.

For the system with 720 asphaltene molecules at 350 ns, the interfacial tension and elasticity was calculated using the previously described methods. The interfacial tension was found to be 39.5 ± 1 mN/m. The elasticity was estimated to be 15 ± 10 mN/m; in Figure 19 the change in tension versus the change in interfacial area is shown together with a line showing the 15 mN/m slope.

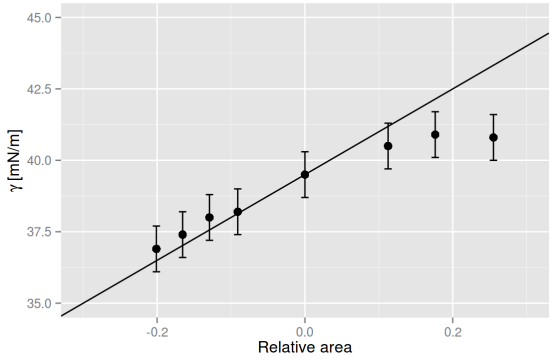



Figure 19: The change in interfacial tension versus the relative area $A/A_0 - 1$ for compressions (negative relative area) and expansions of the interface. It is seen that compressions give a reasonable elastic behaviour, but under expansion the change in tension levels off. The line corresponds to $K_a = 15$ mN/m.

*5.1.2. Studies using the APCE model asphaltene*

As previously mentioned, the new model asphaltenes were developed partly in response to the APCH model asphaltenes being more self-associative than what is expected based on experimental evidence. The APCE asphaltenes adopt a very different configuration at the interface, namely with the polycyclic aromatic core aligned with the water interface, and the aliphatic tails stretched back towards the oil phase. This is shown in Figure 20. In this figure, the interface is seen from the water side, in the system with 240 model asphaltenes after 50 ns of equilibration. In the centre, highlighted by green planes above and below, the interface is shown with the asphaltene molecules as in the previous figures. On the left side, a periodic image is shown with only the aromatic cores. On the right side, a periodic image is shown with only the aliphatic tails. In all three images, grey beads indicate heptane and toluene. It is seen that the aliphatic tails are protruding into the grey oil phase, indicated by the shadows these beads are casting on each other. In Figure 21, the same illustration is shown for the system with 720 asphaltene molecules.

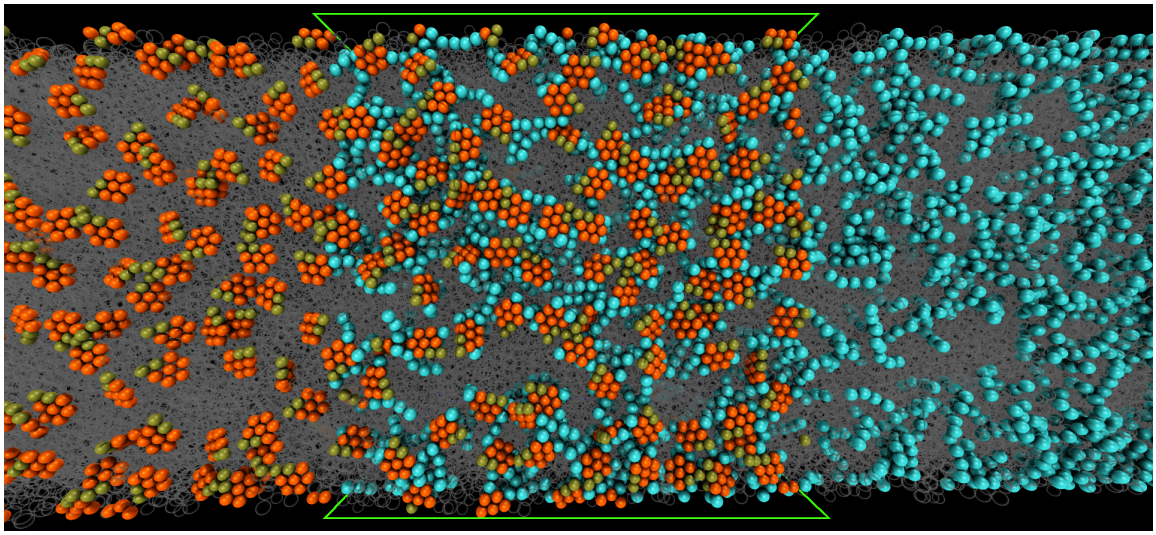


Figure 20: Snapshot of the APCE model asphaltene at the interface seen from the water side, for the system with 240 asphaltenes, after 50 ns of equilibration. Colors as in previous figures. On the right and left sides, periodic images of the interface are shown, where the aromatic cores and the aliphatic tails are omitted, respectively. It is seen that the aromatic cores are predominantly oriented in parallel with the interface, while the aliphatic tails are pointing away into the oil phase.

These results are in very good agreement with experiments performed by Andrews et al. [141] using sum frequency generation (SFG) spectroscopy to study the orientation of real asphaltenes at interfaces. They find that "SFG clearly indicates that asphaltene polycyclic aromatic hydrocarbons are highly oriented in the plane of the interface and that the peripheral alkanes are transverse to the interface." When comparing Figure 17 to Figure 20, it is clear that the APCE model asphaltene has an interfacial behavior closer to that observed in experiments, while results with the APCH model asphaltene show a qualitatively different interfacial orientation. It is noteworthy that previous atomistically detailed molecular simulations [124, 125, 126] have shown similar interfacial behaviour to the APCH model, i.e. have found the asphaltenes to be highly associative at the interface and stacking with cores orthogonal to the interface, in contrast with the experimental findings using real asphaltenes. Thus the results presented here with the APCE model asphaltene are the first simulations to be consistent with the experimental results from SFG spectroscopy [141].

This absence of clustering at the interface means that there is no long time scale for reorientation at the interface, unlike with the APCH asphaltenes. This means that extending simulation runtimes to 350 nanoseconds, as for the APCH asphaltenes, is not necessary. To confirm this, a simulation with the highest number of asphaltene molecules considered here was run until 150 ns, and no change in configuration was

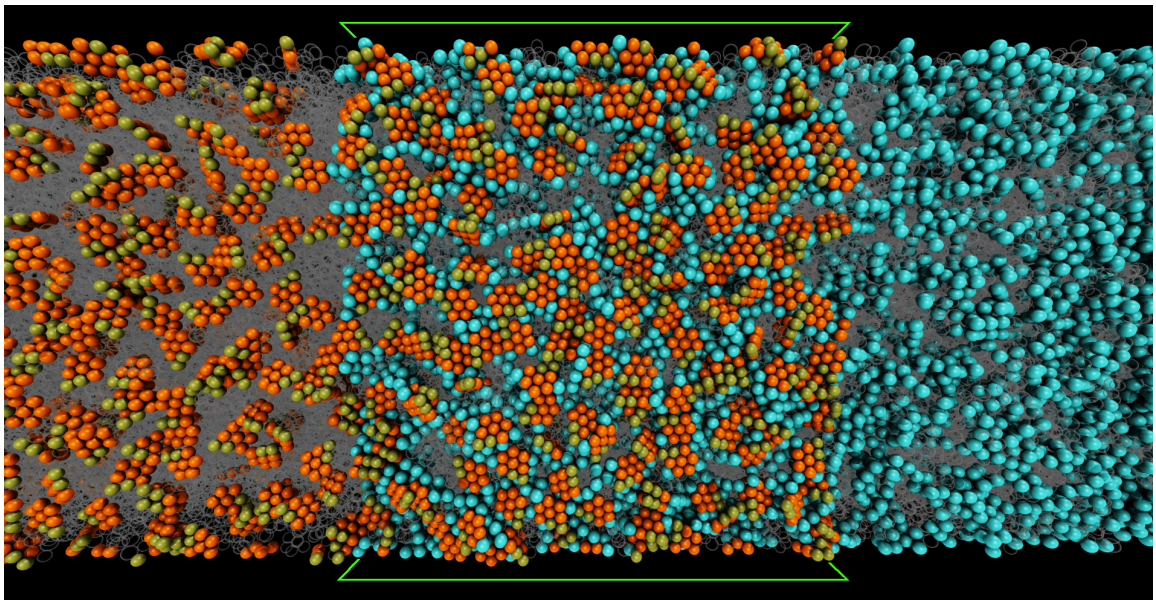


Figure 21: Same figure as Figure 20 for the system with 720 asphaltene molecules. It is seen that even at this high concentration, the asphaltenes show essentially no clustering at the interface. Some asphaltenes here have their cores oriented perpendicularly to the interface, but the great majority still have cores parallel to the interface.

observed between 50 and 150 ns.

With this model giving the interfacial behaviour as expected from experiments, it is interesting to see how the interfacial tension varies with the asphaltene concentration. This is plotted in Figure 22. It is seen that adding asphaltenes to the system decreases the interfacial tension, with what appears to be an exponentially decaying trend. Note that the concentration in this figure is given as the number of molecules, since interpretation of this number into bulk concentration is not immediately obvious. The variation in the number of molecules in the simulation can be interpreted either as snapshots in time during the diffusion process for one bulk concentration, or as the equilibrium interfacial tension for varying concentration of asphaltenes. In general, these values and variation in interfacial tension corresponds well with experimental observations, e.g. in [142].

The interfacial elasticity was also estimated with this model, using the system with 720 molecules after 50 ns, and found to be $55 \pm 20$ mN/m. The variation in tension with the interfacial area is shown in Figure 23, together with a line having a slope of 55 mN/m.

### *5.2. Macroscopic simulations of the draining of water drops in oil*

Given the immense potential for variations in even model crude oil/water systems, and the uncertainties still surrounding the asphaltenes despite the use of state-of-the-art analytic chemistry techniques, it is not possible in the experimental literature to provide enough detail for an exact comparison to be made here between simulations and experiments. It is also noteworthy that the elasticity as defined here has not been given consideration in the experimental literature. There, studies of interfacial properties in crude oil/water systems have either focused on the shear rheology (e.g. [143, 144]), or in the cases where dilatational elasticity has been considered (e.g. [145, 146, 147]), the authors have presupposed this elasticity to be of the type considered by Gibbs in his seminal work [3]. As we will demonstrate in the following, a Gibbs-type elasticity cannot cause the phenomena observed in experiments, for several reasons: Gibbs elasticity is given as the change in interfacial tension when a change in area causes a change in the concentration of a surface-active material. It follows that this elasticity is isotropic by definition, and furthermore that it cannot reduce the

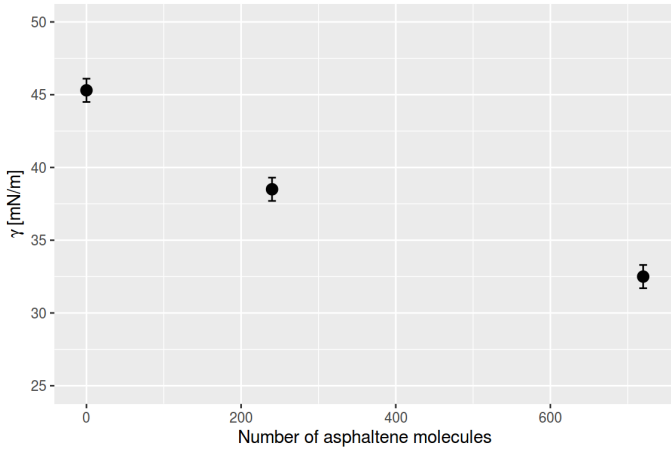



Figure 22: The interfacial tension $\gamma$ computed from molecular simulations, for varying concentration of asphaltenes. The decrease in $\gamma$ with increasing asphaltene concentration corresponds well with experimental measurements.

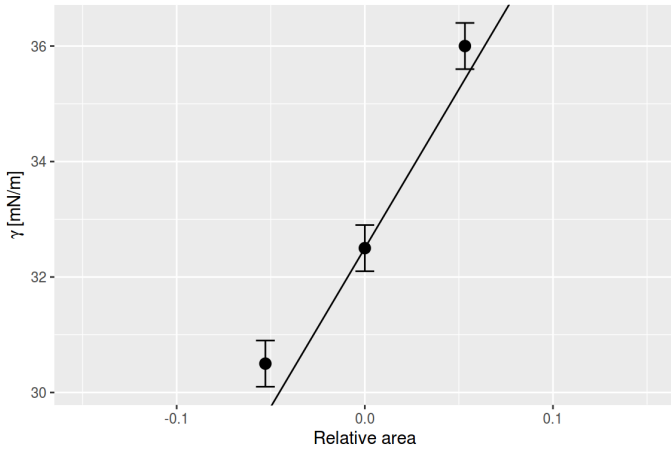



Figure 23: The change in interfacial tension versus the relative area $A/A_0 - 1$ for compression (negative relative area) and expansion of the interface, with the 720 asphaltene molecule system. It is seen that the behaviour is linear, i.e. elastic. The line corresponds to $K_a = 55$ mN/m.

tension to zero. The reader is referred to pp. 467–482 of the very readable paper by Gibbs [3] for further details.

All in all, these facts imply that our comparisons with experiments must be of the qualitative kind. A very interesting case for qualitative comparisons is the crumpled drop phenomenon, which occurs when a water drop in crude oil is drained by use of a needle. The appearance of wrinkles or crumples in the drop interface upon contraction is clearly evident in photographs and is thus not subject to interpretation, and it is a phenomenon fundamentally different from those observed in water/oil/surfactant systems.

In the experimental literature there are two different categories of experiments with draining water drops in crude-oil (or model systems). The first category concerns the micropipette experiments as reported e.g. by [13]. In this case, the drop size is representative of drops found in a real water/crude-oil emulsion, with a drop diameter of 50 $\mu$m. For these drops, buoyancy effects are small when compared to tension effects. In the second category of experiments the pendant drop tensiometer is employed, with much larger drops of

diameter around 5 mm, e.g. as reported by Pauchard et al. [14][3]. In this case, buoyancy effects are larger and tension effects smaller than in the micropipette case.

To get a quantitative measure of this difference, we may consider the Eötvös number, using the inverse drop curvature $1/\kappa$ as the length scale, i.e.

$$Eo = \frac{\Delta\rho g}{T\kappa^2} \tag{44}$$

where $\Delta\rho$ is the water-oil density difference in kg/m$^2$, $g = 9.81$ m/s$^2$ is the gravitational acceleration, and $T$ is the total tension, which is of the order of 0.01 N/m. This means the Eötvös number is $Eo \approx 5 \cdot 10^{-5}$ for the micropipette case, while it is $Eo \approx 0.5$ for the pendant drop case, meaning that in the former case buoyancy is completely negligible, while in the latter case buoyancy has an effect. This means that in the latter case, the interface is deformed both by the action of the tensiometer and by the effect of buoyancy.

*5.2.1. The pendant drop case*

With this in mind, we consider first the situation analogous to the pendant drop tensiometer, where drops are influenced both by buoyancy and by tension effects. In the experiments reported e.g. in [14], the drop remains axisymmetric up to the point of crumpling. It is seen that the crumpling only occurs in the azimuthal direction, thus the interface remains in tension in the meridional direction. This is due to the negative buoyancy of the water drop in oil. The situation can be compared to wrapping a handkerchief around an apple and lifting the corners of the handkerchief together: the cloth will wrinkle in the azimuthal direction, but there will obviously be tension in the meridional direction, since the cloth supports the weight of the apple.

After crumpling has occurred, as long as the crumple depth remains small compared to the drop radius, the axisymmetric simulation remains valid also in the crumpled region [71], where the interface then corresponds to an azimuthal mean interface. This is precisely because the azimuthal tension is zero in the crumpled region, and so this tension (including the effect of azimuthal curvature of the crumples) does not exert a force. Thus axisymmetric simulations can provide insight into this situation. Unfortunately, neither drop diameters prior to deflation, nor the composition (and thus density) of the oil phase, are given in [14]; moreover, the interfacial elasticity is not known, and thus an exact comparison to these experiments cannot be made.

The case considered is a drop of water with an initial diameter of 6.6 mm immersed in oil ($\rho_2 = 830$ kg/m$^3$) and attached to a needle which sucks the water out at a rate of 0.059 mL/s. The initial drop volume is 1.2 mL. The penalisation method is used to enforce no flow inside the needle walls, and a Pouseille flow inside the needle. There is no explicit handling of the drop/needle contact angle, but the interface inside the needle wall is forced to remain in its initial position by the penalisation. The tension in this case is given by $\gamma = 30$ mN/m and $K_a = 50$ mN/m. These specific values are chosen based on the results from the molecular simulations, which have some uncertainty; further details are given in Table A.6. Thus the Eötvös number for the initial condition (where $T = \gamma$) is $Eo = 0.61$, and the simulation corresponds very well to the pendant drop regime.

As the simulation begins, the drop starts to fall, and a balance is quickly established between the drop tension and the drop negative buoyancy, forming a pendant drop. As the needle removes water from the drop, it shrinks, and eventually the "neck" of the drop has shrunk to the point that it becomes tensionless in the azimuthal direction, i.e. $T_\phi = 0$. This is shown in Figure 24.

In this plot, the axisymmetric drop profile is mirrored around the symmetry axis, and the tensions $T_s$ and $T_\phi$ are shown on the right- and left-hand sides, respectively, as colours on the interface. Inside and outside the drop, contours of the pressure field are shown. It is evident that even though $T_\phi$ is zero along a portion of the interface, there is still a pressure jump across the interface. It is also seen that the pressure difference varies along the interface, since $T_\phi$ varies in the meridional direction; and as previously pointed out, there *cannot be* a force which cancels out this variation. This is true even in the static situation.

To compare the simulation result to images from experiments e.g. in [14], it is illustrative to construct a three-dimensional representation from the axisymmetric interfacial profile. To this end, the axisymmetric

[3]The drop diameter is not given in this publication, but the tensiometer needle outer diameter is given as 1.65 mm.

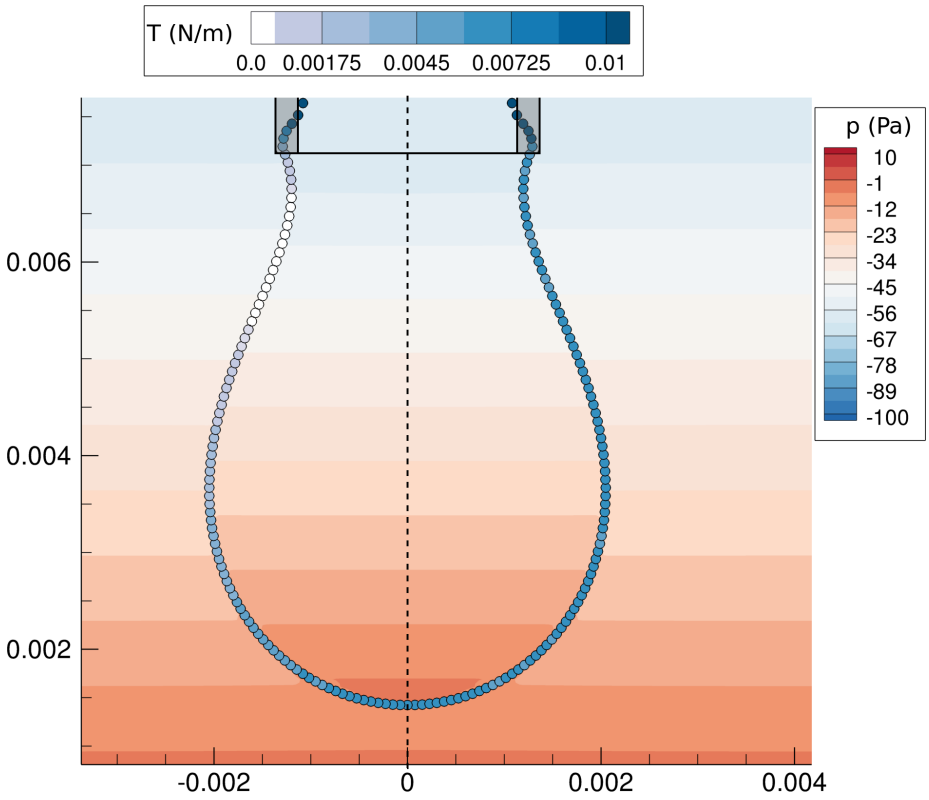



Figure 24: The axisymmetric drop profile (points) mirrored across the symmetry axis (dashed line). Only every 20th Lagrangian point is shown. The colour of the points corresponds to $T_s$ on the right-hand side and $T_\phi$ on the left-hand side of the plot. The pressure is as coloured contours inside and outside the drop, showing the varying pressure jump across the interface.

profile was imported into the Blender 3D graphics software and revolved around the symmetry axis to create a whole drop. Crumples were then inserted manually in the region where $T_\phi = 0$. Raytracing was used to create a realistic rendering of the drop and the needle to which it is attached. The result is shown in Figure 25 next to the experimental result from [14]. The similarity is striking, indicating that elasticity in the interface is a very likely explanation of the crumpled drop phenomenon seen in experiments.

*5.2.2. The micropipette case*

Considering the small Eötvös number regime, analogous to the micropipette drops, the situation after crumpling occurs is clearly not axisymmetric, and thus full three-dimensional simulations are required for quantitative studies of this. While the hybrid method discussed in this paper is readily extendable to three dimensions, this has not been done here due to time constraints, but may be considered in future work. One can, however, study this case from the qualitative perspective by considering a purely two-dimensional simulation, corresponding to a drop and pipette which are significantly elongated in one direction (perpendicular to the simulation domain).

The case considered for this case was with an initial diameter of 2 mm, corresponding to $Eo = 0.06$. While this drop (and $Eo$) is substantially larger than in the experiments, it is small enough that gravity becomes unimportant. Accordingly, the simulation is with zero gravity. Further details of the case are given in Table A.7; see also [90]. A comparison of the simulation result with the photograph from experiments by Yeung et al. [13], both from after crumpling has occurred, is shown in Figure 26. A good qualitative similarity between the two is seen.

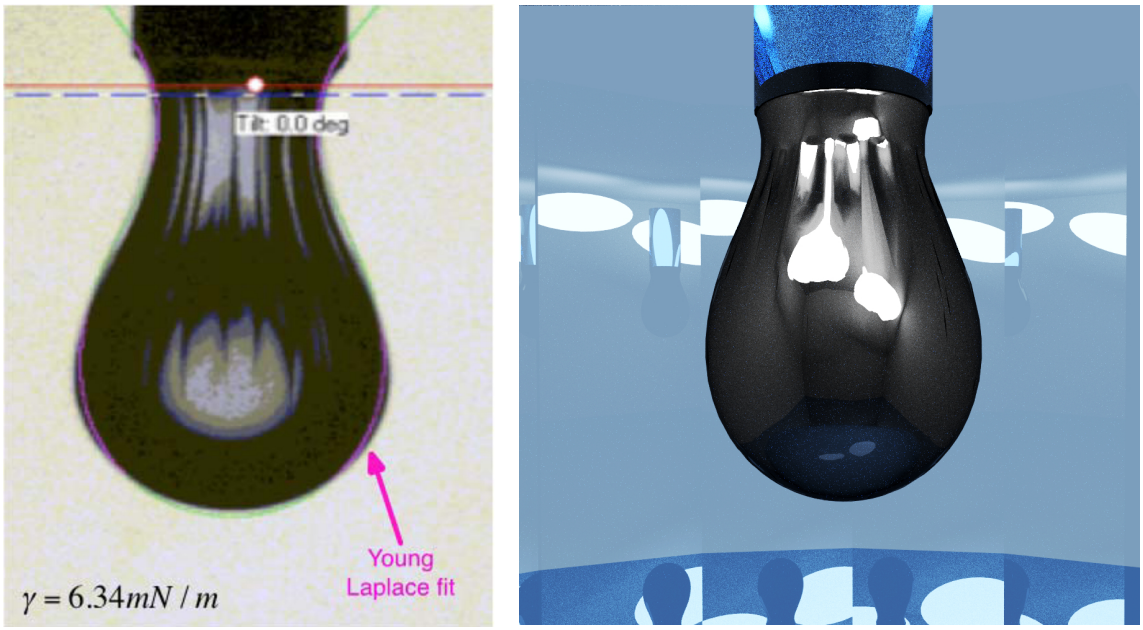



Figure 25: Right: raytrace of the drop profile from simulations, revolved around the symmetry axis and with crumples inserted into the region where $T_\phi = 0$. Left: experimental result showing a deflated asphaltene-stabilised water drop in oil. (Left figure reprinted with permission from: V. Pauchard, J. P. Rane, S. Banerjee, Asphaltene-laden interfaces form soft glassy layers in contraction experiments: A mechanism for coalescence blocking, Langmuir 30 (2014) 12795–12803. Copyright (2014) American Chemical Society.)

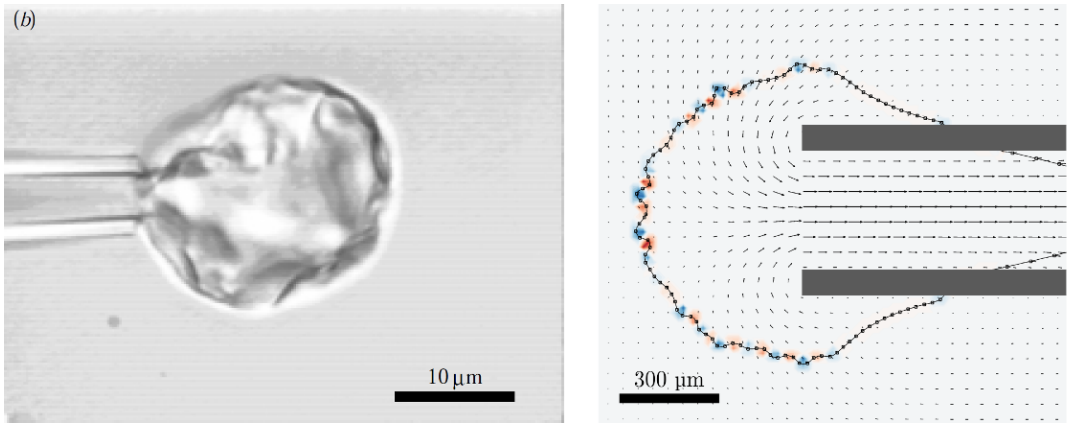



Figure 26: Right: two-dimensional simulation, with red and blue colours indicating interfacial curvature. Velocities are plotted for every 5th grid point and every 10th Lagrangian point is plotted. Left: experimental result showing the crumpled drop attached to a micropipette. (Left figure reprinted with permission from: A. Yeung, T. Dabros, J. Czarnecki, J. Masliyah, On the interfacial properties of micrometre–sized water droplets in crude oil, Proceedings of the Royal Society of London A (1999) 3709–3723.

## 6. Discussion

This paper presents a multiscale approach to simulations that shed light on the behaviour of asphaltene-covered water drops in crude oils. The simulation results at both the molecular and at the continuum scale showcase several interesting phenomena which appear in these systems. Ultimately, a better understanding of these phenomena will lead to a better understanding of crude oil/water emulsions, which are made up of trillions of tiny water drops, each covered with asphaltene molecules. This improved understanding can be utilised to reduce the cost and footprint of oil-water separation equipment, as well as to reduce the use of chemicals and heat in these separators. This in turn leads to reductions in both capital and operational expenses, as well as reduced emissions from oil production.

In the molecular simulations it is found that coarse-grained model asphaltenes are highly sensitive to the molecular architecture, to the point that molecules which are isomers both on the atomistic and on the coarse-grained scales have very different solubilities in the solvents in question (heptane and toluene). This is consistent with e.g. the findings by Sedghi et al. [122] who found in their atomistic simulations that small variations in the side groups of model asphaltenes caused large differences in solubility.

It is notable that the two solvents in question, heptane and toluene, are not very different in the first place: they are miscible, have the same number of carbon atoms, and have similar Hildebrand solubility parameters[4], 15.3 and 18.3 respectively, compared to e.g. 26.2 for ethanol and 48.0 for water, all these from [148] given in $\mathrm{MPa}^{1/2}$. The most obvious difference between heptane and toluene is the different molecular architecture of the two (ring-shaped and linear), which in turn enables toluene to have $\pi$-orbital interactions with other aromatic molecules. With the similarity in mind, an interesting question arises: Asphaltenes are known to be a highly complex mixture of different molecules, which originate as part of the well-stream from crude oil reservoirs deep underground. Unlike the carefully synthesised molecules used in man-made chemical processes, asphaltenes are created in an essentially random fashion. How, then, do they achieve such a detailed solubility balance?

It is likely that the explanation lies in a two-fold application of an argument analogous to the "anthropic principle": the fact that asphaltenes are insoluble in heptane comes from their penchant for $\pi$-$\pi$ ring interactions, which cause them to cluster and subsequently fall out of solution. If the molecules had smaller aromatic cores, they would not show this behaviour, but then they would also not be asphaltenes in the first place, rather they would be resins. The solubility in toluene can be understood from a similar point of view: were the asphaltenes not soluble in toluene (or aromatic solvents in general), they would not be soluble in crude oil at all, so they would never be found in the flow coming up from the reservoir.

An interesting possible topic for investigation in future studies is the effect of asphaltene polydispersity, whic is an obvious next step on the path towards better models for molecular simulations involving asphaltenes. A very appealing idea is the construction of a coarse-grained QMR method, which would in principle enable the construction of an arbitrary number of different coarse-grained model asphaltene molecules.

When the model asphaltenes are introduced into the heptol-water system, they are found to intrinsically be interfacially active. No tuning has been done of the cross-interactions between the aromatic cores and water, due to the lack of experimental properties to tune this interaction to. With this in mind, it is remarkable how well the interfacial behaviour of particularly the APCE model asphaltenes matches what one expects based on our understanding of the $\pi$-interactions between aromatic rings and water [149].

The molecular simulations presented here using the APCE model asphaltenes are the first reported simulations of asphaltenes in oil-water systems which obtain the correct interfacial behaviour, as compared to experimental results using sum frequency generation spectroscopy [141]. This leads one to speculate that perhaps a coarse-grained model asphaltene which has the correct solubility behaviour in heptane and in toluene, such as the APCE model, somehow begins to take the asphaltene polydispersity into account through the coarse-grained nature.

When it comes to the pendant drop case, simulations highlight that the tension is highly anisotropic, and that the azimuthal tension varies significantly along the drop profile, being zero around the neck of the

[4]The Hildebrand solubility parameter is a quantity most useful for comparing solvents, and is given by the square root of the cohesive energy density.

drop when volume has been removed with the needle. This is caused by the Eötvös number being close to one, such that both gravity and the needle suction are responsible for deforming the drop. The resulting variations in the tensions explain the axisymmetric crumpling observed in experiments.

An important implication of the anisotropy in the tension is that when using the pendant drop tensiometer to study asphaltene covered drops, it makes little sense to speak of a "surface pressure" as a scalar number indicative of the total tension in the interface. This is in contrast to the situation with a flat interface, as in a classical Langmuir-Blodgett trough apparatus, where the surface pressure would remain well-defined. This questions the findings in previous studies where conclusions are drawn on the basis of variations in the "surface pressure" of asphaltene-covered drops measured using the pendant drop tensiometer to deform drops. It is also important to note that the elasticity considered in these simulations is fundamentally different from the Gibbs elasticity of surfactants.

When studying ordinary surfactants, the Marangoni effect ensures that (at equilibrium) the distribution of surfactants on the drop interface is uniform, and it thus makes sense to speak of a surface pressure in this case. For the anisotropic tensions studied here, the analogue of the Marangoni effect serves only to remove variation of $T_s$ in the meridional direction etc. But as discussed previously, it is not possible to have forces acting to cancel the variation in $T_\phi$ along the meridional direction, since these forces would be in the direction binormal to the axis of strain.

When it comes to the micropipette drop case, the Eötvös number is $Eo \ll 1$, and thus gravity is unimportant. This means that the tension forces dominate, and that drop deformations are entirely controlled by the needle suction. This causes the drop to remain spherical up to the point of crumpling, such that the crumples appear simultaneously across the drop interface. The two-dimensional simulation at low Eötvös number demonstrates a qualitative similarity to this, in that crumpling appears on all of the interface.

How these macroscopic phenomena relate to emulsion stability remains a topic for future investigations. An important question which should be investigated is the deformation of emulsion drops as they travel from the point of emulsion formation, through the varying turbulence levels in the flow which transports them through the pipeline, and finally when they end up in the oil-water separator, where the flow is less turbulent and the drop shapes will be spherical. Taking this into account, together with the adsorption time scales for asphaltenes in a highly sheared flow, would provide important insight into the situation for real emulsion drops. The multiscale method employed here, with future extensions e.g. to fully three-dimensional flow, is uniquely poised to help answer these and other important questions.

## 7. Concluding remarks

To summarise, the current paper presents a multiscale approach to the simulation of drops with complex interfaces, such as water drops in crude oil which are covered with asphaltene molecules. The approach combines coarse-grained molecular dynamics simulations using the SAFT-$\gamma$ Mie force field, with detailed two-phase flow simulations using a hybrid level-set/ghost-fluid/immersed-boundary method developed as part of this work. At the molecular scale, the coarse-grained approach enables simulations at unprecedented time- and length-scales, using accurate models for both the simple fluids and the complex asphaltene molecules. The interfacial tension $\gamma$ and elasticity $K_a$ are estimated, for use in the macroscopic simulations.

At the macroscopic scale, detailed simulations of oil-water interfaces with both interfacial tension and elastic properties shed new light on experimental results that showcase one of the peculiarities of complex fluid-fluid interfaces, namely crumpling drops. Two categories of experimental results exist in the literature, with different classes of crumpling behaviour. The results presented here demonstrate that this difference is caused by the large difference in Eötvös number between the two categories. In the pendant drop case, with an Eötvös number around one, both gravity and the experimental setup are responsible for deforming the drop, and this combination leads to an axisymmetric crumpling regime. In the micropipette case, with an Eötvös number much smaller than one, the drop remains spherical up to the point of crumpling, leading to a fully three-dimensional behaviour.

These results present a novel hypothesis, namely anisotropic tensions in the interface, as an explanation of the crumpling phenomenon. This hypothesis is different from previously suggested explanations, e.g. that

based on glass transitions in [14]. The hypothesis put forward in this work makes fewer assumptions and has greater predictive power than those presented in previous works.

Building on this work, there are numerous avenues open for future investigations. The effect on interfacial properties of crude oil composition, asphaltene polydispersity, asphaltene architecture, other crude components such as resins, etc., should all be considered. Extending the hybrid simulation method to three dimensions would enable not just simulations of e.g. the micropipette drops, but also of drops in more complicated (even turbulent) flows. Furthermore, the extension to drop coalescence is a very interesting topic. Also, this study assumes that the interfacial elasticity is of the Hookean type. While this is the simplest form of elasticity, it could well be that asphaltene-covered interfaces can be more accurately modelled using e.g. a neo-Hookean elasticity.

All in all, the methods presented here are well suited for increasing our understanding of dispersed two-phase flows with complex interfaces. This is important not just for the application in focus here, crude oil/water systems, but also for biological systems such as the flow of blood or the transport of proteins, and for chemical processes involving multiphase flow and macromolecules.

## Acknowledgements

ÅE, STM and BM acknowledge the financial support from the project *Fundamental understanding of electrocoalescence in heavy crude oils* coordinated by SINTEF Energy Research and funded by the Petromaks programme of the Research Council of Norway (206976), Petrobras, Statoil and Wärtsilä Oil & Gas Systems.

EM acknowledges the financial support for the Molecular Systems Engineering group from the Engineering and Physical Sciences Research Council (grant numbers EP/E016340, EP/J014958, EP/L020564).

The authors would like to thank Prof. Edo Boek for fruitful discussions on these matters. The authors are also grateful for computational resources at the Imperial College High Performance Computing Service, and at the Abel cluster operated by the University of Oslo and the Norwegian metacenter for High Performance Computing (NOTUR), as well as assistance from the staff at both locations.

## References

[1] C. Marangoni, Sull'espansione Delle Gocce di un Liquido Gallegianti Sulla Superficie di Altro Liquido, Ph.D. thesis, University of Pavia, 1865.

[2] C. Marangoni, Über die Ausbreitung der Tropfen einer Flussigkeit auf der Oberfläche einer anderen, Annalen der Physik und Chemie 143 (1870) 337–354.

[3] J. W. Gibbs, On the equlibrium of heterogeneous substances, Transactions of the Connecticut Academy of Arts and Sciences 143 (1878) 337–354.

[4] Pliny, *Naturalis Historia*, Teubner, Leipzig, 1906. http://penelope.uchicago.edu/Thayer/E/Roman/Texts/Pliny_the_Elder/home.html.

[5] Pliny, Pliny's Natural History, Harvard University Press, Massachusetts, 1949. http://www.masseiana.org/pliny.htm.

[6] A. Pockels, Relations between the surface-tension and relative contamination of water surfaces, Nature 48 (1893) 152–154.

[7] B. Szyszkowski, Experimentelle Studien über kapillare Eigenschaften der wässerigen Lösungen von Fettsäuren, Zeitschrift für Physikalische Chemie 64 (1908) 385–414.

[8] I. Langmuir, The adsorption of gases on plane surfaces of glass, mica and platinum., Journal of the American Chemical Society 40 (1918) 1361–1403.

[9] A. Frumkin, On a simple method of testing the Gibbs concepts, Zeitschrift für Physikalische Chemie 116 (1925) 498.

[10] V. Levich, V. Krylov, Surface-tension-driven phenomena, Annual Review of Fluid Mechanics 1 (1969) 293–316.

[11] V. G. Levich, Physicochemical hydrodynamics, Prentice-Hall, 2nd edition, 1962.

[12] T. G. Fai, B. E. Griffith, Y. Mori, C. S. Peskin, Immersed boundary method for variable viscosity and variable density problems using fast constant-coefficient linear solvers i: Numerical method and results, SIAM Journal on Scientific Computing 35 (2013) B1132–B1161.

[13] A. Yeung, T. Dabros, J. Czarnecki, J. Masliyah, On the interfacial properties of micrometre–sized water droplets in crude oil, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 455 (1999) 3709–3723.

[14] V. Pauchard, J. P. Rane, S. Banerjee, Asphaltene-laden interfaces form soft glassy layers in contraction experiments: A mechanism for coalescence blocking, Langmuir 30 (2014) 12795–12803.

[15] M. Boussingault, XCI. On the constitution of bitumens, Philosophical Magazine Series 3 9 (1836) 487–488.

[16] M. P. Hoepfner, Investigations into Asphaltene Deposition, Stability, and Structure, Ph.D. thesis, University of Michigan, 2013.
[17] A. A. Herod, K. D. Bartle, R. Kandiyoti, Comment on a paper by Mullins, Martinez-Haya, and Marshall "Contrasting perspective on asphaltene molecular weight. This comment vs the overview of A. A. Herod, K. D. Bartle, and R. Kandiyoti", Energy & Fuels 22 (2008) 4312–4317.
[18] G. C. Klein, S. Kim, R. P. Rodgers, A. G. Marshall, A. Yen, Mass spectral analysis of asphaltenes. II. Detailed compositional comparison of asphaltenes deposit to its crude oil counterpart for two geographically different crude oils by ESI FT-ICR MS, Energy & Fuels 20 (2006) 1973–1979.
[19] T. Jones, E. Neustadter, K. Whittingham, et al., Water-in-crude oil emulsion stability and emulsion destabilization by chemical demulsifiers, Journal of Canadian Petroleum Technology 17 (1978) –.
[20] O. V. Gafonova, H. W. Yarranton, The stabilization of water-in-hydrocarbon emulsions by asphaltenes and resins, Journal of Colloid and Interface Science 241 (2001) 469–478.
[21] J. Sjöblom (Ed.), Encyclopedic handbook of emulsion technology, CRC Press, 2001.
[22] S. L. Kokal, Crude oil emulsions: A state-of-the-art review., SPE Production & Facilities 20 (2005) 5–13.
[23] P. K. Kilpatrick, Water-in-crude oil emulsion stabilization: Review and unanswered questions, Energy & Fuels 26 (2012) 4017–4026.
[24] C. Dicharry, D. Arla, A. Sinquin, A. Graciaa, P. Bouriat, Stability of water/crude oil emulsions based on interfacial dilatational rheology, Journal of colloid and interface science 297 (2006) 785–791.
[25] J. Bi, F. Yang, D. Harbottle, E. Pensini, P. Tchoukov, S. Simon, J. Sjöblom, T. Dabros, J. Czarnecki, Q. Liu, Z. Xu, Interfacial layer properties of a polyaromatic compound and its role in stabilizing water-in-oil emulsions, Langmuir (2015). Article in Press.
[26] W. E, B. Engquist, Z. Huang, Heterogeneous multiscale method: A general methodology for multiscale modelling, Phys. Rev. B 67 (2003).
[27] A. Brandt, Multiscale and multiresolution methods: Theory and applications, Springer, Berlin, Heidelberg.
[28] S. Chapman, T. G. Cowling, C. Cercignani, The Mathematical Theory of Non-uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases, Cambridge Mathematical Library, 1991.
[29] E. A. Müller, K. E. Gubbins, Molecular-based equations of state for associating fluids: A review of SAFT and related approaches, Industrial & Engineering Chemistry Research 40 (2001) 2193–2211.
[30] I. G. Economou, Statistical associating fluid theory: A successful model for the calculation of thermodynamic and phase equilibrium properties of complex fluid mixtures, Industrial & Engineering Chemistry Research 41 (2002) 953–962.
[31] S. P. Tan, H. Adidharma, M. Radosz, Recent advances and applications of statistical associating fluid theory, Industrial & Engineering Chemistry Research 47 (2008) 8063–8082.
[32] C. McCabe, A. Galindo, SAFT associating fluids and fluid mixtures, Royal Society of Chemistry, London.
[33] T. Lafitte, C. Avendaño, V. Papaioannou, A. Galindo, C. S. Adjiman, G. Jackson, E. A. Müller, SAFT-$\gamma$ force field for the simulation of molecular fluids: 3. Coarse-grained models of benzene and hetero-group models of n-decylbenzene, Molecular Physics 110 (2012) 1189–1203.
[34] V. Papaioannou, T. Lafitte, C. Avendaño, C. S. Adjiman, G. Jackson, E. A. Müller, A. Galindo, Group contribution methodology based on the statistical associating fluid theory for heteronuclear molecules formed from Mie segments, The Journal of Chemical Physics 140 (2014) –.
[35] G. Mie, Zur kinetischen Theorie der einatomigen Körper, Annalen der Physik 316 (1903) 657–697.
[36] N. Ramrattan, C. Avendaño, E. Müller, A. Galindo, A corresponding-states framework for the description of the Mie family of intermolecular potentials, Molecular Physics (2015) 1–16.
[37] W. Noid, Perspective: Coarse-grained models for biomolecular systems, The Journal of chemical physics 139 (2013) 090901.
[38] E. Brini, E. A. Algaer, P. Ganguly, C. Li, F. Rodriguez-Ropero, N. F. A. van der Vegt, Systematic coarse-graining methods for soft matter simulations - a review, Soft Matter 9 (2013) 2108–2119.
[39] S. J. Marrink, H. J. Risselada, S. Yefimov, D. P. Tieleman, A. H. De Vries, The MARTINI force field: coarse grained model for biomolecular simulations, The Journal of Physical Chemistry B 111 (2007) 7812–7824.
[40] X. Periole, S.-J. Marrink, The martini coarse-grained force field, Biomolecular Simulations: Methods and Protocols (2013) 533–565.
[41] E. A. Müller, A. Mejía, Resolving discrepancies in the measurements of the interfacial tension for the CO2 + H2O mixture by computer simulation, The Journal of Physical Chemistry Letters 5 (2014) 1267–1271.
[42] C. Avendaño, T. Lafitte, A. Galindo, C. S. Adjiman, G. Jackson, E. A. Müller, SAFT-$\gamma$ force field for the simulation of molecular fluids. 1. A single-site coarse grained model of carbon dioxide, The Journal of Physical Chemistry B 115 (2011) 11154–11169.
[43] C. Avendaño, T. Lafitte, C. S. Adjiman, A. Galindo, E. A. Müller, G. Jackson, SAFT-$\gamma$ force field for the simulation of molecular fluids: 2. Coarse-grained models of greenhouse gases, refrigerants, and long alkanes, The Journal of Physical Chemistry B 117 (2013) 2717–2733.
[44] O. Lobanova, C. Avendaño, T. Lafitte, E. A. Müller, G. Jackson, SAFT-$\gamma$ force field for the simulation of molecular fluids: 4. A single-site coarse-grained model of water applicable over a wide temperature range, Molecular Physics 113 (2015) 1228–1249.
[45] A. Mejía, C. Herdes, E. A. Müller, Force fields for coarse-grained molecular simulations from a corresponding states correlation, Industrial & Engineering Chemistry Research 53 (2014) 4131–4141.
[46] C. Herdes, E. Forte, G. Jackson, E. Müller, Predicting the adsorption of n-perfluorohexane in BAM P109 standard activated carbon by molecular simulation using SAFT-$\gamma$ Mie coarse-grained force fields, Adsorption Science and Technology

33 (2015) –.
[47] C. Herdes, E. Santiso, C. James, J. Eastoe, E. Müller, Modelling the interfacial behaviour of dilute light-switching surfactant solutions, Journal of Colloid and Interface Science 445 (2015) 16–23.
[48] P. E. Theodorakis, E. A. Müller, R. V. Craster, O. K. Matar, Superspreading: Mechanisms and molecular design, Langmuir 31 (2015) 2304–2309.
[49] E. A. Müller, C. Herdes, T. S. Totton, Coarse-grained models for crude oils: A direct link between equations of state and molecular simulations, in: Offshore Technology Conference.
[50] M. P. Allen, D. J. Tildesley, Computer simulation of liquids, Oxford university press, 1989.
[51] D. Frenkel, B. Smit, Understanding molecular simulation: from algorithms to applications, volume 1, Academic press, 2001.
[52] B. J. Alder, T. E. Wainwright, Phase transition for a hard sphere system, The Journal of Chemical Physics 27 (1957) 1208–1209.
[53] B. J. Alder, T. E. Wainwright, Molecular motions, Scientific American 201 (1959) 113–126.
[54] K. Kadau, T. C. Germann, P. S. Lomdahl, Molecular dynamics comes of age: 320 billion atom simulation on BlueGene/L, International Journal of Modern Physics C 17 (2006) 1755–1761.
[55] G. Zhao, J. R. Perilla, E. L. Yufenyuy, X. Meng, B. Chen, J. Ning, J. Ahn, A. M. Gronenborn, K. Schulten, C. Aiken, et al., Mature HIV-1 capsid structure by cryo-electron microscopy and all-atom molecular dynamics, Nature 497 (2013) 643–646.
[56] Å. Ervik, G. J. Serratos, E. A. Müller, raaSAFT: a framework enabling coarse-grained molecular dynamics simulations based on the SAFT-$\gamma$ Mie force field, Submitted (2016).
[57] J. A. Anderson, C. D. Lorenz, A. Travesset, General purpose molecular dynamics simulations fully implemented on graphics processing units, Journal of Computational Physics 227 (2008) 5342 – 5359.
[58] J. Glaser, T. D. Nguyen, J. A. Anderson, P. Lui, F. Spiga, J. A. Millan, D. C. Morse, S. C. Glotzer, Strong scaling of general-purpose molecular dynamics simulations on GPUs, Computer Physics Communications 192 (2015) 97 – 107.
[59] G. J. Martyna, D. J. Tobias, M. L. Klein, Constant pressure molecular dynamics algorithms, The Journal of Chemical Physics 101 (1994) 4177–4189.
[60] W. G. Hoover, J. C. Sprott, P. K. Patra, Ergodic time-reversible chaos for Gibbs' canonical oscillator, Physics Letters A 379 (2015) 2935 – 2940.
[61] P. K. Patra, J. C. Sprott, W. G. Hoover, C. G. Hoover, Deterministic time-reversible thermostats: chaos, ergodicity, and the zeroth law of thermodynamics, Molecular Physics 113 (2015) 2863–2872.
[62] W. G. Hoover, An appreciation of Berni Julian Alder, arXiv preprint 1510.05897 (2015).
[63] J. G. Kirkwood, F. P. Buff, The statistical mechanical theory of surface tension, The Journal of Chemical Physics 17 (1949) 338–343.
[64] C. Vega, E. de Miguel, Surface tension of the most popular models of water by using the test-area simulation method, The Journal of Chemical Physics 126 (2007) –.
[65] W. K. den Otter, W. J. Briels, The bending rigidity of an amphiphilic bilayer from equilibrium and nonequilibrium molecular dynamics, The Journal of Chemical Physics 118 (2003) 4712–4720.
[66] E. S. Boek, J. T. Padding, W. K. den Otter, W. J. Briels, Mechanical properties of surfactant bilayer membranes from atomistic and coarse-grained molecular dynamics simulations, The Journal of Physical Chemistry B 109 (2005) 19851–19858.
[67] G. Batchelor, An Introduction to Fluid Dynamics, Cambridge Mathematical Library, Cambridge University Press, 2000.
[68] H. Lamb, Hydrodynamics, Dover, New York, 6 edition, 1945.
[69] G. V. Lau, I. J. Ford, P. A. Hunt, E. A. Müller, G. Jackson, Surface thermodynamics of planar, cylindrical, and spherical vapour-liquid interfaces of water, The Journal of Chemical Physics 142 (2015).
[70] C.-Y. Hui, A. Jagota, Surface tension, surface energy, and chemical potential due to their difference, Langmuir 29 (2013) 11310–11316.
[71] S. Knoche, J. Kierfeld, Buckling of spherical capsules, Phys. Rev. E 84 (2011) 046608.
[72] C. S. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.
[73] S. Knoche, D. Vella, E. Aumaitre, P. Degen, H. Rehage, P. Cicuta, J. Kierfeld, Elastometry of deflated capsules: Elastic moduli from shape and wrinkle analysis, Langmuir 29 (2013) 12463–12471.
[74] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202 – 228.
[75] G.-S. Jiang, D. Peng, Weighted ENO schemes for Hamilton-Jacobi equations, SIAM J. Sci. Comput. 21 (2000) 2126–2143.
[76] A. J. Chorin, Numerical solution of the Navier-Stokes equations, Math. Comput. 22 (1968) 745–762.
[77] V. E. Henson, U. M. Yang, BoomerAMG: a parallel algebraic multigrid solver and preconditioner, Appl. Numer. Math. 41 (2000) 155–177.
[78] H. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Comput. 13 (1992) 631–644.
[79] R. Falgout, J. Jones, U. Yang, The Design and Implementation of hypre, a Library of Parallel High Performance Preconditioners, Springer-Verlag, pp. 267–294.
[80] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, Efficient management of parallelism in object-oriented numerical software libraries, in: A. Bruaset, E. Arge, H. Langtangen (Eds.), Modern Software Tools for Scientific Computing, Springer, 1997, pp. 163–202.
[81] M. Kang, R. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, Journal of Scientific Computing 15 (2000) 323–360.
[82] S. Gottlieb, D. I. Ketcheson, C.-W. Shu, High order strong stability preserving time discretizations, J. Sci. Comput. 38

(2009) 251–289.
[83] J. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows, Comput. Meth. Appl. Mech. Eng. 195 (2006) 6011 – 6045.
[84] K. E. Teigen, S. T. Munkejord, Sharp-interface simulations of drop deformation in electric fields, IEEE T. Dielect. El. In. 16 (2009) 475–482.
[85] K. E. Teigen, K. Y. Lervåg, S. T. Munkejord, Sharp interface simulations of surfactant-covered drops in electric fields, in: Fifth European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2010.
[86] K. E. Teigen, S. T. Munkejord, Influence of surfactant on drop deformation in an electric field, Phys. Fluids 22 (2010) –.
[87] Å. Ervik, K. Y. Lervåg, S. T. Munkejord, A robust method for calculating interface curvature and normal vectors using an extracted local level set, Journal of Computational Physics 257, Part A (2014) 259 – 277.
[88] Å. Ervik, S. M. Hellesø, S. T. Munkejord, B. Müller, Experimental and computational studies of water drops falling through model oil with surfactant and subjected to an electric field, in: Proceedings of the IEEE 18th International Conference on Dielectric Liquids, Bled, Slovenia.
[89] Å. Ervik, T. E. Penne, S. M. Hellesø, S. T. Munkejord, B. Müller, Influence of surfactants on the electrohydrodynamic stretching of water drops in oil, Submitted (2016).
[90] M. O. Lysgaard, Development of an immersed boundary method for simulating contaminated fluid interfaces in two-phase flow, Master's thesis, Department of Mathematics, Norwegian University of Science and Technology, 2015.
[91] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, Journal of Computational Physics 79 (1988) 12 – 49.
[92] D. Adalsteinsson, J. Sethian, A fast level set method for propagating interfaces, Journal of Computational Physics 118 (1995) 269–277. Cited By 0.
[93] V. Sabelnikov, A. Y. Ovsyannikov, M. Gorokhovski, Modified level set equation and its numerical assessment, Journal of Computational Physics 278 (2014) 1 – 30.
[94] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Applied Mathematical Sciences, Springer New York, 2012.
[95] R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[96] R. J. LeVeque, Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput. 18 (1997) 709–735.
[97] J. Brackbill, D. B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335–354.
[98] C. Peskin, Numerical analysis of blood flow in the heart, Journal of Computational Physics 25 (1977) 220–252. Cited By 1276.
[99] K. Y. Lervåg, Calculation of interface curvatures with the level-set method for two-phase flow simulations and a second-order diffuse-domain method for elliptic problems in complex geometries, Ph.D. thesis, Norwegian University of Science and Technology, Faculty of Engineering Science and Technology, Department of Energy and Process Engineering, 2013.
[100] M. Sussman, E. G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, Journal of Computational Physics 162 (2000) 301–337.
[101] M. Francois, W. Shyy, Computations of drop dynamics with the immersed boundary method, part 1: numerical algorithm and buoyancy-induced effect, Numerical Heat Transfer, Part B: Fundamentals 44 (2003) 101–118.
[102] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, Journal of Computational Physics 209 (2005) 448–476.
[103] C. Tu, C. S. Peskin, Stability and instability in the computation of flows with moving immersed boundaries: A comparison of three methods, SIAM Journal on Scientific and Statistical Computing 13 (1992) 1361–1376.
[104] C. Peskin, D. McQueen, A general method for the computer simulation of biological systems interacting with fluids, in: Symposia of the society for Experimental Biology, volume 49, pp. 265–276. Cited By 58.
[105] J. Lucassen, Longitudinal capillary waves. Part 1. Theory, Transactions of the Faraday Society 64 (1968) 2221–2229.
[106] J. Lucassen, Longitudinal capillary waves. Part 2. Experiments, Transactions of the Faraday Society 64 (1968) 2230–2235.
[107] P. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, Numerische Mathematik 81 (1999) 497–520.
[108] M. Kang, R. P. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, Journal of Scientific Computing 15 (2000) 323–360.
[109] R. J. Leveque, High-resolution conservative algorithms for advection in incompressible flow, SIAM Journal on Numerical Analysis 33 (1996) pp. 627–665.
[110] S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, Journal of Computational Physics 31 (1979) 335 – 362.
[111] O. Lobanova, A. Mejía, G. Jackson, E. A. Müller, SAFT-$\gamma$ force field for the simulation of molecular fluids 6: Binary and ternary mixtures comprising water, carbon dioxide, and n-alkanes, The Journal of Chemical Thermodynamics 93 (2016) 320 – 336.
[112] J. Saien, , S. Akbari, Interfacial tension of toluene + water + sodium dodecyl sulfate from 20 to 50 °C and pH between 4 and 9, Journal of Chemical & Engineering Data 51 (2006) 1832–1835.
[113] H. W. Yarranton, J. H. Masliyah, Gibbs-Langmuir model for interfacial tension of nonideal organic mixtures over water, The Journal of Physical Chemistry 100 (1996) 1786–1792.
[114] S. Zeppieri, J. Rodríguez, A. L. L. de Ramos, Interfacial tension of alkane + water systems, Journal of Chemical & Engineering Data 46 (2001) 1086–1088.

[115] E. Rogel, L. Carbognani, Density estimation of asphaltenes using molecular dynamics simulations, Energy & fuels 17 (2003) 378–386.
[116] J. Pacheco-Sánchez, I. Zaragoza, J. Martínez-Magadán, Asphaltene aggregation under vacuum at different temperatures by molecular dynamics, Energy & fuels 17 (2003) 1346–1355.
[117] S. R. Stoyanov, S. Gusarov, A. Kovalenko, Multiscale modelling of asphaltene disaggregation, Molecular Simulation 34 (2008) 953–960.
[118] T. Kuznicki, J. H. Masliyah, S. Bhattacharjee, Aggregation and partitioning of model asphaltenes at toluene- water interfaces: Molecular dynamics simulations, Energy & Fuels 23 (2009) 5027–5035.
[119] T. F. Headen, E. S. Boek, N. T. Skipper, Evidence for asphaltene nanoaggregation in toluene and heptane from molecular dynamics simulations, Energy & Fuels 23 (2009) 1220–1229.
[120] E. S. Boek, D. S. Yakovlev, T. F. Headen, Quantitative molecular representation of asphaltenes and molecular dynamics simulation of their aggregation, Energy & Fuels 23 (2009) 1209–1219.
[121] T. F. Headen, E. S. Boek, Molecular dynamics simulations of asphaltene aggregation in supercritical carbon dioxide with and without limonene, Energy & Fuels 25 (2010) 503–508.
[122] M. Sedghi, L. Goual, W. Welch, J. Kubelka, Effect of asphaltene structure on association and aggregation using molecular dynamics, The Journal of Physical Chemistry B 117 (2013) 5765–5776.
[123] P. Ungerer, D. Rigby, B. Leblanc, M. Yiannourakou, Sensitivity of the aggregation behaviour of asphaltenes to molecular weight and structure using molecular dynamics, Molecular Simulation 40 (2014) 115–122.
[124] Y. Mikami, Y. Liang, T. Matsuoka, E. S. Boek, Molecular dynamics simulations of asphaltenes at the oil–water interface: from nanoaggregation to thin-film formation, Energy & Fuels 27 (2013) 1838–1845.
[125] J. Liu, Y. Zhao, S. Ren, Molecular dynamics simulation of self-aggregation of asphaltenes at an oil/water interface: formation and destruction of the asphaltene protective film, Energy & Fuels 29 (2015) 1233–1242.
[126] F. Yang, P. Tchoukov, H. Dettman, R. B. Teklebrhan, L. Liu, T. Dabros, J. Czarnecki, J. Masliyah, Z. Xu, Asphaltene subfractions responsible for stabilizing water-in-crude oil emulsions. Part 2: Molecular representations and molecular dynamics simulations, Energy & Fuels 29 (2015) 4783–4794.
[127] Y. Ruiz-Morales, O. C. Mullins, Coarse-grained molecular simulations to investigate asphaltenes at the oil–water interface, Energy & Fuels 29 (2015) 1597–1609.
[128] D. Fenistein, L. Barre, Experimental measurement of the mass distribution of petroleum asphaltene aggregates using ultracentrifugation and small-angle X-ray scattering, Fuel 80 (2001) 283–287.
[129] N. H. Rahmani, T. Dabros, J. H. Masliyah, Fractal structure of asphaltene aggregates, Journal of colloid and interface science 285 (2005) 599–608.
[130] N. V. Lisitza, D. E. Freed, P. N. Sen, Y.-Q. Song, Study of asphaltene nanoaggregation by nuclear magnetic resonance (NMR), Energy & Fuels 23 (2009) 1189–1193.
[131] O. C. Mullins, H. Sabbah, J. Eyssautier, A. E. Pomerantz, L. Barré, A. B. Andrews, Y. Ruiz-Morales, F. Mostowfi, R. McFarlane, L. Goual, et al., Advances in asphaltene science and the Yen–Mullins model, Energy & Fuels 26 (2012) 3986–4003.
[132] J. D. McLean, P. K. Kilpatrick, Effects of asphaltene aggregation in model heptane-toluene mixtures on stability of water-in-oil emulsions, Journal of Colloid and Interface Science 196 (1997) 23 – 34.
[133] J. Djuve, X. Yang, J. I. Fjellanger, J. Sjöblom, E. Pelizzetti, Chemical destabilization of crude oil based emulsions and asphaltene stabilized emulsions, Colloid and Polymer Science 279 (2001) 232–239.
[134] P. Spiecker, K. L. Gawrys, C. B. Trail, P. K. Kilpatrick, Effects of petroleum resins on asphaltene aggregation and water-in-oil emulsion formation, Colloids and Surfaces A: Physicochemical and Engineering Aspects 220 (2003) 9 – 27.
[135] P. Spiecker, K. L. Gawrys, P. K. Kilpatrick, Aggregation and solubility behavior of asphaltenes and their subfractions, Journal of Colloid and Interface Science 267 (2003) 178 – 193.
[136] P. M. Spiecker, P. K. Kilpatrick, Interfacial rheology of petroleum asphaltenes at the oil-water interface, Langmuir 20 (2004) 4022–4032.
[137] H. Yarranton, D. Sztukowski, P. Urrutia, Effect of interfacial rheology on model emulsion coalescence: I. Interfacial rheology, Journal of Colloid and Interface Science 310 (2007) 246 – 252.
[138] S. Wang, J. Liu, L. Zhang, J. Masliyah, Z. Xu, Interaction forces between asphaltene surfaces in organic solvents, Langmuir 26 (2010) 183–190.
[139] D. M. Sztukowski, H. W. Yarranton, Rheology of asphaltene-toluene/water interfaces, Langmuir 21 (2005) 11651–11658.
[140] K. Norinaga, V. J. Wargardalam, S. Takasugi, M. Iino, , S. Matsukawa, Measurement of self-diffusion coefficient of asphaltene in pyridine by pulsed field gradient spin-echo 1H NMR, Energy & Fuels 15 (2001) 1317–1318.
[141] A. B. Andrews, A. McClelland, O. Korkeila, A. Demidov, A. Krummel, O. C. Mullins, Z. Chen, Molecular orientation of asphaltenes and PAH model compounds in langmuir-blodgett films using sum frequency generation spectroscopy, Langmuir 27 (2011) 6049–6058.
[142] M. Fossen, H. Kallevik, K. D. Knudsen, J. Sjöblom, Asphaltenes precipitated by a two-step precipitation procedure. 1. Interfacial tension and solvent properties, Energy & Fuels 21 (2007) 1030–1037.
[143] V. J. Verruto, R. K. Le, P. K. Kilpatrick, Adsorption and molecular rearrangement of amphoteric species at oil-water interfaces, The Journal of Physical Chemistry B 113 (2009) 13788–13799.
[144] Y. Fan, S. Simon, J. Sjöblom, Interfacial shear rheology of asphaltenes at oil–water interface and its relation to emulsion stability: influence of concentration, solvent aromaticity and nonionic surfactant, Colloids and Surfaces A: Physicochemical and Engineering Aspects 366 (2010) 120–128.
[145] K. Kumar, A. Nikolov, D. Wasan, Mechanisms of stabilization of water-in-crude oil emulsions, Industrial & engineering chemistry research 40 (2001) 3009–3014.

[146] N. Aske, R. Orr, J. Sjöblom, Dilatational elasticity moduli of water–crude oil interfaces using the oscillating pendant drop, Journal of dispersion science and technology 23 (2002) 809–825.
[147] D. Daniel-David, I. Pezron, C. Dalmazzone, C. Noïk, D. Clausse, L. Komunjer, Elastic properties of crude oil/water interface in presence of polymeric emulsion breakers, Colloids and Surfaces A: Physicochemical and Engineering Aspects 270 (2005) 257–262.
[148] A. F. Barton, Handbook of Solubility Parameters, CRC Press, 1983.
[149] S. Li, V. R. Cooper, T. Thonhauser, A. Puzder, D. C. Langreth, A density functional theory study of the benzene-water complex, The Journal of Physical Chemistry A 112 (2008) 9031–9036.

## Appendix A. Parameters for the continuum simulation cases

Table A.1: Parameters for the drop in potential vortex.

| Parameter | Symbol | Value |
|---|---|---|
| Drop radius | $r$ | 0.15 |
| Domain size | $\Omega$ | $1 \times 1$ |
| Stability safety factor | $C$ | 0.5 |
| Euler grid nodes | $N$ | $200 \times 200$ |
| Lagrangian point density | | $5/\Delta$ |

Table A.2: Parameters for the Zalesak's disk test.

| Parameter | Symbol | Value |
|---|---|---|
| Disk radius | $r$ | $1/3$ |
| Domain size | $\Omega$ | $1 \times 1$ |
| Time step safety factor | $C$ | 0.5 |
| Euler grid nodes | $N$ | $64 \times 64$ |
| Velocity field | $\mathbf{u}$ | $\frac{\pi}{10}[1/2 - y, x - 1/2]$ |
| Lagrangian point density | | $5/\Delta$ |

Table A.3: Parameters for the elliptical drop driven by interfacial tension.

| Parameter | Symbol | Value |
|---|---|---|
| Drop density | $\rho_1$ | $10^3\,\mathrm{kg/m^3}$ |
| Matrix density | $\rho_2$ | $10^3\,\mathrm{kg/m^3}$ |
| Drop viscosity | $\mu_1$ | $10^{-3}\,\mathrm{Pa \cdot s}$ |
| Matrix viscosity | $\mu_2$ | $10^{-3}\,\mathrm{Pa \cdot s}$ |
| Interfacial tension | $\gamma$ | $15 \times 10^{-3}\,\mathrm{N/m}$ |
| Drop radius | $r$ | $10^{-3}\,\mathrm{m}$ |
| Drop axis length ratio | $\frac{a}{b}$ | 1.16 |
| Domain size | $\Omega$ | $0.007 \times 0.007\,\mathrm{m}$ |
| Time step safety factor | $C$ | 0.2 |
| Grid nodes | $N$ | $\{100, 200, 400, 800\}$ |

Table A.4: Parameters for relaxing drop with viscosity and density jump.

| Parameter | Symbol | Value |
|---|---|---|
| Drop density | $\rho_1$ | $10^3\,\mathrm{kg/m^3}$ |
| Matrix density | $\rho_2$ | $5 \times 10^2\,\mathrm{kg/m^3}$ |
| Drop viscosity | $\mu_1$ | $10^{-3}\,\mathrm{Pa \cdot s}$ |
| Matrix viscosity | $\mu_2$ | $10^{-2}\,\mathrm{Pa \cdot s}$ |
| Interfacial tension | $\gamma$ | $15 \times 10^{-3}\,\mathrm{N/m}$ |
| Drop radius | $r$ | $10^{-3}\,\mathrm{m}$ |
| Drop axis length ratio | $\frac{a}{b}$ | 1.16 |
| Domain size | $\Omega$ | $0.007 \times 0.007\,\mathrm{m}$ |
| Time step safety factor | $C$ | 0.2 |
| Grid nodes | $N$ | 400 |

Table A.5: Parameters for relaxing drop with an elastic membrane.

| Parameter | Symbol | Value |
|---|---|---|
| Drop density | $\rho_1$ | $10^3\,\mathrm{kg/m^3}$ |
| Matrix density | $\rho_2$ | $10^3\,\mathrm{kg/m^3}$ |
| Drop viscosity | $\mu_1$ | $10^{-3}\,\mathrm{Pa \cdot s}$ |
| Matrix viscosity | $\mu_2$ | $10^{-3}\,\mathrm{Pa \cdot s}$ |
| Interfacial tension | $\gamma$ | $15 \times 10^{-3}\,\mathrm{N/m}$ |
| Elasticity | $K_a$ | 0 and $15 \times 10^{-2}\,\mathrm{N/m}$ |
| Drop radius | $r$ | $10^{-3}\,\mathrm{m}$ |
| Drop axis length ratio | $\frac{a}{b}$ | 3.0 |
| Domain size | $\Omega$ | $0.007 \times 0.007\,\mathrm{m}$ |
| Time step safety factor | $C$ | 0.5 |
| Grid nodes | $N$ | 150 |

Table A.6: Parameters for the pendant drop case.

| Parameter | Symbol | Value |
|---|---|---|
| Drop density | $\rho_1$ | $1000\,\mathrm{kg/m^3}$ |
| Matrix density | $\rho_2$ | $830\,\mathrm{kg/m^3}$ |
| Drop viscosity | $\mu_1$ | $1.03 \times 10^{-3}\,\mathrm{Pa \cdot s}$ |
| Matrix viscosity | $\mu_2$ | $12.4 \times 10^{-3}\,\mathrm{Pa \cdot s}$ |
| Interfacial tension | $\gamma$ | $30 \times 10^{-3}\,\mathrm{N/m}$ |
| Elasticity | $K_a$ | $50 \times 10^{-3}\,\mathrm{N/m}$ |
| Drop radius | $r$ | $3.3 \times 10^{-3}\,\mathrm{m}$ |
| Domain size | $\Omega$ | $(5 \times 10^{-3}) \times (15 \times 10^{-3})\,\mathrm{m}$ |
| Time step safety factor | $C$ | 0.3 |
| Grid nodes | $N$ | $132 \times 200$ |
| Penalisation | $\eta$ | $3 \times 10^{-5}$ |

Table A.7: Parameters for the micropipette case.

| Parameter | Symbol | Value |
|---|---|---|
| Drop density | $\rho_1$ | $1000\,\mathrm{kg/m^3}$ |
| Matrix density | $\rho_2$ | $830\,\mathrm{kg/m^3}$ |
| Drop viscosity | $\mu_1$ | $1.03 \times 10^{-3}\,\mathrm{Pa \cdot s}$ |
| Matrix viscosity | $\mu_2$ | $12.4 \times 10^{-3}\,\mathrm{Pa \cdot s}$ |
| Interfacial tension | $\gamma$ | $40 \times 10^{-3}\,\mathrm{N/m}$ |
| Elasticity | $K_a$ | $50 \times 10^{-3}\,\mathrm{N/m}$ |
| Drop radius | $r$ | $1 \times 10^{-3}\,\mathrm{m}$ |
| Domain size | $\Omega$ | $(2 \times 10^{-3}) \times (3 \times 10^{-3})\,\mathrm{m}$ |
| Time step safety factor | $C$ | 0.5 |
| Grid nodes | $N$ | $132 \times 200$ |
| Penalisation | $\eta$ | $5 \times 10^{-6}$ |

## Appendix B. Parameters and scripts for the molecular simulations

The molecular simulations reported here are performed using our raaSAFT framework. Scripts which can be used to reproduce the molecular simulations are available in the `replication` directory of the raaSAFT repository, at `http://www.bitbucket.org/asmunder/raasaft`. The force field parameters are mostly published elsewhere, as indicated by the DOI for each model in that repository. However, the force field parameters for the aromatic beads used in the APCE and APCL asphaltene models have not been published elsewhere yet. These are given in Table B.8. The parameters for the toluene model are also not published elsewhere yet, these are given in Table B.8.

Table B.8: Parameters for the aromatic beads.

| Parameter | Value | Unit |
|---|---|---|
| | **Toluene** | |
| $\epsilon/k_B$ | 269.74 | [K] |
| $\sigma$ | 3.6794 | [Å] |
| $n$ | 11.804 | [-] |
| $m$ | 6 | [-] |
| | **APCE** | |
| $\epsilon/k_B$ | 312.90 | [K] |
| $\sigma$ | 3.975 | [Å] |
| $n$ | 10 | [-] |
| $m$ | 6 | [-] |