

Numerical modeling of turbulent dense dispersed two-phase flows

Kwakkel, Marcel

DOI

[10.4233/uuid:9c32f850-1179-45ff-ba37-cd8e441b4795](https://doi.org/10.4233/uuid:9c32f850-1179-45ff-ba37-cd8e441b4795)

Publication date

2016

Document Version

Final published version

Citation (APA)

Kwakkel, M. (2016). *Numerical modeling of turbulent dense dispersed two-phase flows*. [Dissertation (TU Delft), Delft University of Technology]. <https://doi.org/10.4233/uuid:9c32f850-1179-45ff-ba37-cd8e441b4795>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

NUMERICAL MODELING OF TURBULENT
DENSE DISPERSED TWO-PHASE FLOWS



NUMERICAL MODELING OF TURBULENT DENSE DISPERSED TWO-PHASE FLOWS

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben;
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
vrijdag 10 juni 2016 om 10:00 uur

door

Marcel KWAKKEL
ingenieur Luchtvaart en Ruimtevaart
Technische Universiteit Delft
geboren te Heerde.

This dissertation has been approved by the

promotor: Prof. dr. ir. B.J. Boersma

copromotor: Dr. ir. W.P. Breugem

Composition of the doctoral committee:

Rector Magnificus

Prof. dr. ir. B.J. Boersma

Dr. ir. W.P. Breugem

chairman

Delft University of Technology

Delft University of Technology

Independent members:

Prof. dr. ir. R.A.W.M. Henkes

Prof. dr. ir. N.G. Deen

Prof. dr. ir. C. Vuik

Prof. dr. J.G.M. Kuerten

Dr. ir. M.I. Gerritsma

Delft University of Technology

Eindhoven University of Technology

Delft University of Technology

Eindhoven University of Technology

Delft University of Technology

Copyright © 2016 by M. Kwakkel

All rights reserved.

ISBN 978-94-6186-659-2

Printed by Ipskamp Printing, Enschede

CONTENTS

SUMMARY	vii
SAMENVATTING	ix
1 INTRODUCTION	1
1.1 Turbulent dispersed two-phase flows	1
1.2 Characteristics of turbulent dispersed two-phase flows	4
1.3 Research approaches for turbulent dispersed two-phase flows	7
1.3.1 Governing equations for interfacial two-phase flows	10
1.3.2 Dimensionless parameters	15
1.4 Research objectives	16
1.5 Thesis outline	18
2 A MORE EFFICIENT MULTIPLE MARKER CLSVOF METHOD	21
2.1 Introduction	21
2.2 Multiple marker front-capturing method	23
2.2.1 Governing equations	23
2.2.2 Discretization of the Navier-Stokes equations	24
2.2.3 Interface representation	25
2.2.4 Surface tension	26
2.2.5 Parallel implementation	27
2.2.6 Conjugate Gradient method	28
2.3 Numerical improvements	29
2.3.1 Local multiple marker method	29
2.3.2 Deflation method	30
2.3.3 Three-dimensional domain decomposition	33
2.4 Numerical experiments	33
2.4.1 Effect of local multiple marker method	35
2.4.2 Effect of deflation method	37
2.4.3 Effect of three-dimensional domain decomposition	38
2.4.4 Overall effect	41
2.5 Conclusions	42
3 EXTENSION WITH A COALESCENCE/BREAKUP MODEL	45
3.1 Introduction	45
3.2 Original numerical method	48
3.2.1 Governing equations	48
3.2.2 Time integration and spatial discretization	50
3.2.3 Multiple marker CLSVOF method	51
3.2.4 Parallelization	53
3.3 Extension with coalescence/breakup model	53

Contents

3.3.1	Coalescence model	53
3.3.2	Breakup model	59
3.3.3	Numerical implementation	60
3.4	Numerical results	65
3.4.1	Collision dynamics based on observed drainage times	66
3.4.2	Effect of film drainage time on collision dynamics	70
3.4.3	Effect of grid resolution on collision dynamics	72
3.5	Conclusions and discussion	74
4	TURBULENT BUBBLY DOWNFLOW IN A VERTICAL CHANNEL	79
4.1	Introduction	79
4.2	Numerical method	82
4.2.1	Governing equations	82
4.2.2	Flow geometry and scaling	83
4.3	Turbulent single-phase channel flow	87
4.3.1	Flow initialization	87
4.3.2	Turbulence statistics	89
4.4	Turbulent bubbly two-phase channel flow	95
4.4.1	Bubbly flow specification	95
4.4.2	Bubbly flow initialization	97
4.4.3	Preliminary results	97
4.5	Computational costs	102
4.5.1	Computational resources	102
4.5.2	Time step size	103
4.5.3	Time step breakdown	104
4.6	Conclusions and outlook	110
5	MAIN CONCLUSIONS AND OUTLOOK	113
5.1	Main conclusions	113
5.1.1	Efficiency improvements	113
5.1.2	Coalescence/breakup modeling	114
5.1.3	Large scale simulations	115
5.2	Outlook	117
	BIBLIOGRAPHY	119
	ACKNOWLEDGEMENTS	133
	ABOUT THE AUTHOR	135

Numerical modeling of turbulent dense dispersed two-phase flows

Turbulent dense dispersed two-phase flows are frequently encountered in natural and industrial processes. Detailed understanding of the dynamics and interaction of the individual phases in these processes is important for their optimization. Such flows can be characterized by mutual influence between the phases, and frequent topological changes. However, these complexities limit the usability of theoretical and experimental methods to obtain quantitative data. Although computationally expensive, Direct Numerical Simulation (DNS) of interfacial two-phase flow can provide the desired physical insight, and data for development and validation of models.

To be able to perform DNS of a large number of interface, an accurate and efficient method is required. The work described in this thesis has been performed to develop and validate such a numerical tool, capable of simulating $\mathcal{O}(10^3)$ dispersed droplets in a turbulent carrier fluid. Application of the multiple marker Coupled Level-Set and Volume-of-Fluid (CLSVOF) method provides sufficient accuracy. In conventional Level-Set and Volume-of-Fluid methods, when two interfaces are less than one grid cell apart from each other, so-called numerical coalescence occurs. In the multiple marker CLSVOF method each interface has its own locally defined marker functions, which prevents this problem. The motion of both fluid phases is computed on a fixed grid, where each fluid phase has its own density and viscosity. At the moving and deformable interfaces between the phases the effect of surface tension and jump conditions for the discontinuous material properties are incorporated in the equations for fluid motion.

Preliminary simulations revealed that the efficiency was mainly compromised by sequential advection of interfaces, and solving the variable coefficient Poisson equation for the pressure. After the implementation of a parallel interface advection algorithm, which is based on a master/slave technique, the method scales well with the number of interfaces. By addition of a second preconditioner to the preconditioned Conjugate Gradient solver, which is based on the deflation technique, the most unfavorable eigenvalues are removed and overall time spend in the pressure solver decreases. Furthermore, to maximize parallelization benefits, the one-dimensional domain decomposition has been extended to a three-dimensional domain decomposition.

Since topological changes are likely to occur in turbulent dense dispersed two-phase flows, the multiple marker CLSVOF method has been extended with a physical model to accommodate coalescence and breakup of droplets. In the model coalescence is based on a computationally efficient film drainage model, which predicts if and when two colliding droplets will coalesce. To validate the proposed coalescence model, simulations of (near) head-on collisions of two equal droplets at four different Weber numbers, corresponding to four different collision regimes,

have been performed. A good agreement with available experimental data for each Weber number has demonstrated the capability of the CLSVOF method to simulate coalescence and breakup with physical realism. A sensitivity analysis has shown a strong influence of the prescribed film drainage time on the collision dynamics, while the influence of the spatial grid resolution appeared to be small. This results has underlined the importance of accurate models for the film drainage time.

To investigate the applicability of the method to more challenging flows, it has been applied to a turbulent bubble-laden vertical channel downflow. Such a setup allows the study of the mutual interaction between the bubbles and the turbulent carrier fluid. To be able to quantify the influence of the (coalescence and breakup of) bubbles on the turbulent flow, a staged strategy has been chosen. First, a turbulent single-phase downflow has been simulated and analyzed, which can be used as a reference for subsequent two-phase flow simulations. Next, 384 bubbles have been added, placed in a slightly perturbed regular array configuration to enhance initial bubble interactions. Thanks to the multiple marker technique, the computational cost of the bubble advection are small. However, during the two-phase flow simulation two major computational deficiencies have been encountered, that together significantly increased the overall computational time. First, the time spend in the iterative pressure solver during the two-phase flow simulation is much higher than expected, probably caused by a less effective removal of unfavorable eigenvalues by the deflation method. Second, although the flow field remained divergence free, occasionally non-physically high velocities inside some bubbles have been observed. As a result, in order to satisfy the CFL time step criterion for numerical stability, the computational time step is reduced considerably. These observed high velocities appear to originate from inconsistencies in the discretization schemes for advective mass and momentum transport, which emerge especially when the mass density ratio is very large. Due to these accumulating computational issues it has been decided that simulation of bubble-laden flows currently is not feasible within reasonable time with the DNS code developed in this project.

It can be concluded that the developed multiple marker CLSVOF method can be used to accurately study coalescence and breakup effects. However, due the encountered computational deficiencies, the current implementation is not optimal to study turbulent dispersed two-phase vertical channel flows in full detail. In recent literature a few methods have been proposed to improve the efficiency of the pressure solver, and ensure consistent advective mass and momentum transport. Therefore, it is believed that the multiple marker CLSVOF method can still provide an efficient and accurate way to study dense dispersions including coalescence and breakup effects.

Numerieke modellering van turbulente dichtgepakte gedispergeerde tweefase stromingen

Turbulente dichtgepakte gedispergeerde tweefase stromingen worden veelvuldig aangetroffen in natuurlijke en industriële processen. Gedetailleerd inzicht in de dynamiek en interactie van de afzonderlijke fasen van deze processen is van belang voor hun optimalisering. Dergelijke stromingen kunnen gekenmerkt worden door wederzijdse beïnvloeding tussen de fasen, en frequente topologische veranderingen. Echter, deze complexiteiten beperken de toepasbaarheid van theoretische en experimentele methoden voor het verkrijgen van kwantitatieve gegevens. Hoewel computationeel duur, kan Directe Numerieke Simulatie (DNS) van tweefase stromingen met een grensvlak het gewenste fysische inzicht, en gegevens voor de ontwikkeling en validatie van modellen leveren.

Om DNS van een groot aantal grensvlakken te kunnen uitvoeren is een nauwkeurige en efficiënte methode vereist. Het werk beschreven in dit proefschrift is uitgevoerd om een numerieke tool te ontwikkelen en valideren, welke in staat is $\mathcal{O}(10^3)$ gedispergeerde druppels in een turbulente dragervloeistof te simuleren. Toepassing van de meervoudige marker 'Coupled Level-Set and Volume-of-Fluid' (CLSVOF) methode levert voldoende nauwkeurigheid. In tegenstelling tot conventionele 'Level-Set' en 'Volume-of-Fluid' methoden, heeft in de meervoudige marker CLSVOF methode elk grensvlak zijn eigen lokaal gedefinieerde marker functies, wat zogenaamde numerieke coalscentie voorkomt. De vloeistofstroming van beide fasen wordt berekend op een stationair rooster, waarbij elke fase zijn eigen dichtheid en viscositeit heeft. Ter plaatse van het bewegende en vervormbare grensvlak tussen de fasen worden het effect van oppervlaktespanning en de sprong voorwaarden voor de discontinue materiaaleigenschappen in de vergelijkingen voor de vloeistofstroming opgenomen.

Voorafgaande simulaties hebben laten zien dat de efficiëntie vooral beïnvloed werd door een seriële advection van grensvlakken, en het oplossen van de variabele coëfficiënt Poisson vergelijking voor de druk. Na implementatie van een parallel grensvlak advection algoritme schaalde de methode goed met het aantal grensvlakken. Door toevoeging van een tweede preconditioner aan de gepreconditioneerde Conjugate Gradient solver, die is gebaseerd op de techniek deflatie, worden de ongunstigste eigenwaarden verwijderd en neemt de totale tijd in de druk solver af. Bovendien, om parallelisatievoordelen te maximaliseren, is de ééndimensionale domein decompositie uitgebreid tot een driedimensionale domein decompositie.

Aangezien topologische veranderingen zeer waarschijnlijk optreden in turbulente dichtgepakte gedispergeerde tweefase stromingen, is de meervoudige marker CLSVOF methode uitgebreid met een fysisch model dat voorziet in coalescentie en opbreken van druppels. Het coalescentiemodel is gebaseerd op een computationeel efficiënt leeglopend film model, dat voorspelt of en wanneer twee botsende drup-

pels zullen coalesceren. Om het voorgestelde coalescentiemodel te valideren zijn simulaties van (bijna) frontale botsingen van twee gelijke druppels met vier verschillende Webergetallen, overeenstemmend met vier verschillende botsingsregimes, uitgevoerd. Een goede overeenkomst met de beschikbare experimentele gegevens voor elk Webergetal heeft aangetoond dat de CLSVOF methode in staat is om coalescentie en opbreken met fysieke realisme te simuleren. Een gevoeligheidsanalyse heeft een sterke invloed van de voorgeschreven film leeglooptijd op de botsingsdynamiek aangetoond, terwijl de invloed van de ruimtelijke grid resolutie klein bleek te zijn. Deze resultaten onderstrepen het belang van nauwkeurige modellen voor de film leeglooptijd.

Om de toepasbaarheid van de methode op meer uitdagende stromingen te onderzoeken, is deze toegepast op een neerwaartse turbulente kanaalstroming gevuld met bellen. Deze opzet maakt het mogelijk om de wisselwerking tussen de bellen en de turbulente dragervloeistof te bestuderen. Om in staat te zijn de invloed van (coalescentie en opbreken van) bellen op de turbulente stroming te kwantificeren, is er voor een gefaseerde strategie gekozen. Eerst is er een turbulent éénfasige neerwaartse stroming gesimuleerd en geanalyseerd, die gebruikt kan worden als referentie voor navolgende simulaties van tweefasestromingen. Vervolgens zijn er 384 bellen toegevoegd, welke in een enigszins verstoorde regelmatige gridconfiguratie zijn geplaatst om initiële bel interacties te versterken.

Dankzij de meervoudige marker techniek zijn de computationele kosten van de advection van bellen klein. Echter, tijdens de tweefasestroming simulatie zijn er twee belangrijke computationele tekortkomingen ondervonden, welke samen de totale rekentijd significant verhogen. Ten eerste kost de iteratieve druk solver in de tweefasestroming simulatie meer tijd dan verwacht, waarschijnlijk veroorzaakt door een minder effectieve verwijdering van ongunstige eigenwaarden door de deflatie methode. Ten tweede, hoewel het stromingsveld divergentievrij bleef, zijn er van tijd tot tijd niet-fysieke hoge snelheden binnen enkele bellen waargenomen. Tengevolge, om te voldoen aan het CFL tijdstap criterium voor numerieke stabiliteit, wordt de computationele tijdstap flink verkleind. Deze hoge snelheden lijken afkomstig te zijn van inconsistenties in de discretisatie schema's voor advection massa en impuls transport, die vooral optreden bij een grote massadichtheidsverhouding. Als gevolg van deze accumulerende computationele problemen is besloten dat simulatie van stromingen gevuld met bellen op dit moment niet haalbaar is binnen een redelijke termijn met de DNS code ontwikkeld in dit project.

Er kan geconcludeerd worden dat de ontwikkelde meervoudige marker CLSVOF methode gebruikt kan worden voor het nauwkeurig bestuderen van coalescentie en opbrek effecten. Echter, vanwege de ondervonden computationele tekortkomingen, is de huidige implementatie niet optimaal voor het bestuderen van turbulente gedispergeerde tweefase stromingen in een verticaal kanaal in detail. In recente literatuur zijn een aantal methoden voorgesteld om de efficiëntie van de druk solver te verbeteren, en te zorgen voor een consistent advection massa en impuls transport. Het is daarom te verwachten dat de meervoudige marker CLSVOF methode nog altijd een efficiënt en nauwkeurig manier kan bieden om dichtgepakte dispersies met coalescentie en opbrek effecten te bestuderen.

INTRODUCTION

Every accomplishment starts with a decision to try

— Gail Devers

1.1 TURBULENT DISPERSED TWO-PHASE FLOWS

Multiphase flows are characterized by the existence of interfaces between different phases of matter, where physical properties are discontinuous at the interfaces and uniform within each phase. In the field of fluid dynamics, the study of these multiphase flows is of continuous interest because of its relevance to many environmental, natural, and industrial processes involving such flows. Examples can be found e.g. in nuclear reactors, production of natural gas, combustion processes, emulsification in food industry, and rain formation in clouds. Understanding of the dynamics and interaction of the various phases in these processes is important for their optimization, i.e. maximizing efficiency, while minimizing costs and environmental impact.

Although the number of phases can be large, most multiphase flows are essentially two-phase flows. Common classifications of two-phase flows are found in terms of the physical state of the phases, or in terms of the structure of the interfaces separating the phases. Based on the physical state of the different phases, two-phase flows can be subdivided in gas-liquid, gas-solid, and liquid-solid flows. The differences between these physical states are mainly characterized by the material properties. Furthermore, liquids/solids may be regarded as incompressible, while gases are generally compressible. Solids are usually undeformable, and have no internal motion. Finally, gas/liquid interfaces can become contaminated, which changes the interfacial tension. Throughout this work the terms bubble, droplet, and fluid particle are interchangeable. However, phases always remain incompressible, interfaces between phases are deformable, and do not become contaminated. Whenever two fluids with different physical properties flow simultaneously through a channel, there is a wide range of possible flow patterns/regimes, see Fig. 1.1. In bubbly/droplet flows, small volumes of gas/liquid are dispersed in a continuous liquid/gas phase. Slug, froth churn, and annular droplet flows are all examples of separated flows, where the phases are spatially disassociated from each other. In between these extremes, several transient flows can be observed.

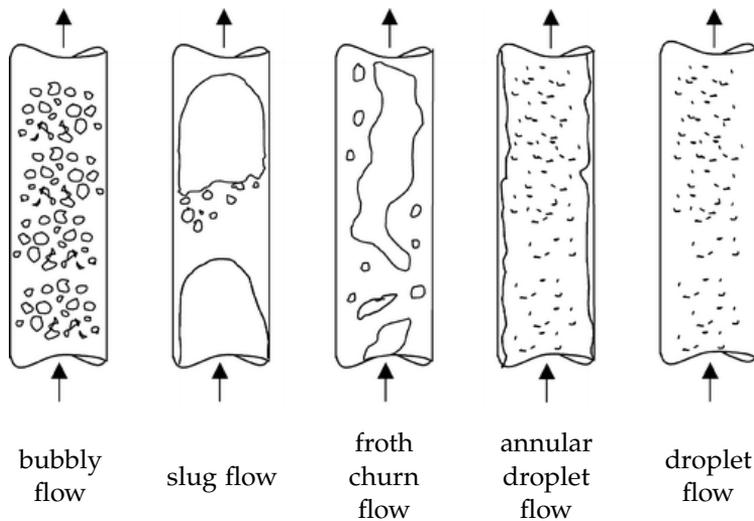


Figure 1.1: Vertical gas-liquid flow patterns according to Hewitt and Hall-Taylor [58].

Due to the wide spectrum of classifications of two-phase flows, this research deals only with incompressible dispersed two-phase fluid flows. Although this can be perceived as a limitation, the regular occurrence of such flows in real life processes does ensure its relevance. Some examples of processes in which two-phase flows play an important role are shown in Fig. 1.2. Fig. 1.2a shows water waves at the ocean surface, which enhance the inclusion of air bubbles in the water phase. These air bubbles increase the absorption rate of carbon dioxide (CO_2) by the ocean. Due to the rising concentration of CO_2 in the Earth's atmosphere, and the acidification of the ocean as a result, understanding of this process is necessary. Fig. 1.2b shows the bubble trails produced by emperor penguins, which are believed to reduce their drag [29]. Understanding of this phenomenon can be beneficial for its application on ships to reduce skin-friction drag [53]. Fig. 1.2c shows the aeration of water, which can be used to increase the oxygen content of water to house animals, treatment of sewage or industrial wastewater, but also to smooth and reduce the flow of tap water at a faucet. Fig. 1.2d shows a medical application, where the droplet size and velocity influence the location where the medicine is deposited. Suboptimal depositing will decrease the effectiveness of the treatment. Improved understanding of the physics involved in all these examples is important to be able to control their functioning, which enables more efficiency and/or less environmental impact.



(a) Water waves at the ocean surface increase the absorption rate of carbon dioxide by the ocean. Image source: www.wallpaperist.com.



(b) Bubble trails of emperor penguins, which are believed to reduce drag [29]. Photograph by Paul Nicklen.



(c) To increase the oxygen level in water, air is injected. Smaller bubbles are thereby more efficient. Image source: www.stoneycreekequip.com.



(d) The effectiveness of nasal spray depends in part on the size of the droplets. Image source: www.flickr.com/photos/robin24.

Figure 1.2: Examples of processes in which two-phase flows play an important role.

Of key interest are in particular fluid particle laden turbulent two-phase flows, where a large number of fluid particles is dispersed in a turbulent carrier fluid. When the dispersed phase volume fraction and the number of fluid particles are high, strong mutual interactions between fluid particles and the turbulent carrier fluid arise. First, the turbulent carrier fluid will influence the movement and thereby size distribution of fluid particles. Furthermore, coalescence and breakup events, which directly influence the size distribution, will occur more often. The fluid particle size distribution is indicative for the amount of interfacial area. Since interfacial area generally determines heat and mass transfer rates, size distributions provide essential information on the efficiency of fluid particle laden turbulent two-phase flows. Second, the fluid particles will influence the structure and dynamics of the turbulent carrier fluid, which might lead to skin-friction drag reduction [29, 53].

1.2 CHARACTERISTICS OF TURBULENT DISPERSED TWO-PHASE FLOWS

Turbulent dispersed two-phase flows can be characterized by various parameters, which provide a basis for the way such flows can be modeled. In this section some of these parameters are explained in more detail.

An important assumption at the base of many flow equations is the continuum hypothesis. Every substance consists of discrete molecules, which collisions determine the macroscopic properties of the substance. If these molecules can freely move through space, the substance is defined to be a fluid (liquid or gas), otherwise to be a solid. Since collisions directly determine these macroscopic properties, the collision frequency divided by a certain reference collision frequency is a good measure for the importance of individual collisions on the macroscopic properties. The collision frequency is inversely proportional to the mean free path of a molecule, which is the average distance molecules travel between collisions. In a similar way, the reference collision frequency is related to the size of the (reference) system, characterized by a length scale of observation [60]. The fraction of the mean free path of a molecule, λ , and the representative physical length scale, L , is known as the dimensionless Knudsen number

$$Kn = \frac{\lambda}{L}. \quad (1.1)$$

In general, three flow regimes based on Kn can be distinguished, see Table 1.1. A more detailed discussion on this subject can be found in the work of Hoitinga [60] and references therein. The continuum hypothesis assumes that a substance

Table 1.1: Flow regimes associated with the Knudsen number [60].

	Scale regime	Flow regime	Equations
$Kn \lesssim 0.01$	Macroscale	Continuum	Navier-Stokes equations
$0.01 \lesssim Kn \lesssim 10$	Mesoscale	Transitional	Boltzmann equations
$Kn \gtrsim 10$	Microscale	Free molecular	Molecular dynamics

is sufficiently dense that averaging over a large number of molecules, over vanishingly small regions, permits a meaningful definition of macroscopic quantities, see Fig. 1.3. Of course this approach has certain inherent restrictions, and these may be expressed in terms of the Knudsen number. An example can be found during the collision of two fluid particles, where the characteristic physical length scale is the film thickness h . Before the fluid particles coalesce, this film thickness can reach a value of $\mathcal{O}(10)$ nm or less [166]. Since in a gas at 1 atm λ is about $\mathcal{O}(10^2)$ nm, Kn can be as large as $\mathcal{O}(10)$. From Table 1.1 it is clear that during the drainage of the film a wide range of flow regimes occur. Therefore, during the drainage of the film the continuum approximation becomes invalid, and rarefied gas effects should be taken into account [166].

One way to classify dispersed two-phase flows is by defining it as dense or dilute. In general the division between dense and dilute dispersed phase flow is based on the controlling forces for the dispersed phase motion. A mixture is dilute if this

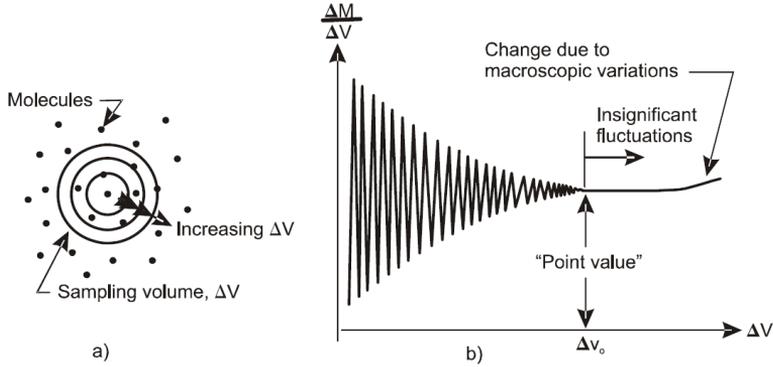


Figure 1.3: The variation of molecular mass per unit volume of mixture with size of sampling volume [28].

motion is controlled by fluid forces (like lift and drag), and dense if collision/contact forces are important. To quantify this division, the volume fraction of the dispersed phase, α_d , and the fluid particle spacing l/D can be used. Here l is a measure of the spacing between two fluid particles of diameter D . A total dispersed phase volume of n spherical dispersed fluid particles with diameter D in a total volume \mathcal{V}_t gives

$$\alpha_d = \frac{n\mathcal{V}_d}{\mathcal{V}_t} = \frac{n\pi D^3}{6\mathcal{V}_t}. \quad (1.2)$$

Combining this with the relations $\mathcal{V}_t = \mathcal{V}_c + \mathcal{V}_d = n l^3$ (where c and d denote the continuous and dispersed phase, respectively), the fluid particle spacing l/D is given by

$$l/D = \left(\frac{\pi}{6\alpha_d} \right)^{1/3}. \quad (1.3)$$

If this spacing is $\mathcal{O}(10)$, the influence of dispersed particles on each other can not be neglected and the system is considered dense. Elghobashi [38] considered suspensions with $\alpha_d < 10^{-3}$ as dilute, and with $\alpha_d > 10^{-3}$ as dense.

Another important factor in dispersed flows is the characteristic time required for fluid particles to adjust to changes in the continuous phase flow. This effect is defined by the dimensionless Stokes number, St , which is the ratio of the (fluid) particle response time scale and the continuous flow time scale

$$St = \frac{\tau_p}{\tau_f}. \quad (1.4)$$

Fig. 1.4 graphically shows the Stokes number effect on the motion of a dispersed (fluid) particle. If $St \ll 1$, the (fluid) particles respond much faster than the flow changes, so these particles can be treated as perfect tracer particles. On the other hand, if $St \gg 1$, the (fluid) particles respond very slowly to changes in the flow velocity. These particles will not be affected much by changes in the flow, and

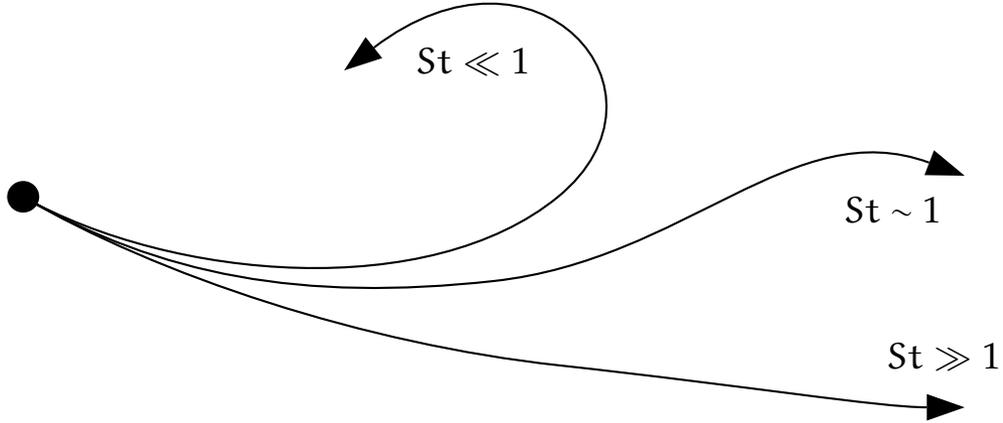


Figure 1.4: Stokes number effect on the motion of a dispersed fluid particle.

will persist in their own motion. At $St \sim 1$, turbulent eddies induce significant coherent motion of the (fluid) particles, which results in the strongest preferential concentration. Lighter-than-fluid particles tend to accumulate in regions of high vorticity, where heavier-than-fluid particles tend to accumulate in regions of high strain rate [6]. [161]. The fluid particle response time, τ_p , can be characterized by analyzing the force balance of a fluid particle, see Crowe et al. [28, Chapter 4] for more details. If a stationary continuous flow is considered, and only drag and virtual mass forces are taken into account, one obtains

$$m \frac{d\mathbf{v}}{dt} = \frac{1}{2} C_D \frac{\pi D^2}{4} \rho_c (\mathbf{u} - \mathbf{v}) |\mathbf{u} - \mathbf{v}|. \quad (1.5)$$

Here $m = (\rho_d + \rho_c C_a) \mathcal{V}_p$ is the virtual mass of the fluid particle, \mathbf{v} is its velocity, and \mathbf{u} the surrounding continuous phase velocity. For a spherical particle in free space the added mass coefficient C_a is equal to 1/2. The fluid particle Reynolds number, Re_p , is defined by

$$Re_p = \frac{\rho_c D |\mathbf{u} - \mathbf{v}|}{\mu_c}. \quad (1.6)$$

Combining these relations gives

$$\frac{d\mathbf{v}}{dt} = \underbrace{\frac{36\mu_c}{(2\rho_d + \rho_c)D^2}}_I \underbrace{\frac{C_D Re_p}{24}}_{II} (\mathbf{u} - \mathbf{v}). \quad (1.7)$$

The drag coefficient C_D for a solid spherical particle in the Stokes limit ($Re_p \ll 1$) is $24/Re_p$, therefore the second term approaches unity. The first term is the inverse response time of the (fluid) particle, which for a solid particle in a gaseous medium becomes

$$\tau_p = \frac{\rho_d}{18\mu_c} D^2. \quad (1.8)$$

Different time scales can be used for the flow time scale, which depends on the flow characteristics. A characteristic time scale for the macroscopic motion of the

carrier phase in a channel configuration is given by $\tau_b = L/U_b$, where L is a characteristic dimension of the channel, and U_b is the bulk velocity [156]. This forms the macroscopic Stokes number $St_b = \tau_p/\tau_b$. For turbulent wall bounded flows a common choice is the viscous time scale $\tau^+ = \nu_c/u_\tau^2$, where u_τ is the so-called wall friction velocity and can be related to the square root of the mean wall shear stress, which is a time scale expressed in wall units. This time scale is relevant in the near-wall region, and is used to form $St^+ = \tau_p/\tau^+$. A characteristic time scale for the smallest eddies in a turbulent carrier flow is given by the Kolmogorov time scale $\tau_\eta = \sqrt{\nu_c/\varepsilon}$, where ε is the viscous dissipation rate, which forms $St_\eta = \tau_p/\tau_\eta$.

Finally, an important aspect in two-phase flows is the way interactions between the two phases are handled. There can be mass exchange (evaporation, condensation), momentum exchange, and energy exchange (heat transfer) between the phases. Since in this work both mass and energy transfer are neglected, here only momentum coupling is described. If and how both phases influence each other in two-phase flows can be determined by the dispersed phase volume fraction, mass loading, and the spacing L/D between dispersed fluid particles. If the effects of the dispersed phase on the continuous phase and its turbulence properties can be neglected, there is a one-way momentum coupling. This is in general the case for dilute flows, where the dispersed phase volume fraction is small, the dispersed phase density is much smaller than the continuous phase density, and $L/D \rightarrow \infty$. However, if the continuous phase and its turbulence properties are influenced by the dispersed phase, there will be momentum transfer in both directions. This regime is therefore denoted as two-way momentum coupling. Lastly, if the dispersed phase volume fraction is even higher and $L/D \rightarrow 1$, there is also momentum transfer within the dispersed phase. Due to interactions between dispersed phase fluid particles, this regime is denoted as four-way momentum coupling.

1.3 RESEARCH APPROACHES FOR TURBULENT DISPERSED TWO-PHASE FLOWS

Improved knowledge of turbulent dispersed two-phase flows can be gained by theoretical, experimental or numerical research. Due to the complexity of turbulent dispersed two-phase flows, the utility of theoretical methods is only possible with many assumptions (simplifications) and thereby very limited. Advantages of experimental research are that all flow scales are present and insight in the phenomenology can be obtained. Furthermore, experimental methods can provide accurate point measurements by using hot-film and hot-wire anemometry, which have a very high time resolution. However, disadvantages are the low spatial resolution and the influence of the probe (support) on the flow. Particle image velocimetry (PIV) can provide unsteady and multidimensional velocity fields by following the motion of tracer particles (which are assumed not to influence the flow). However, application of PIV is limited to two-phase flows with low volume fractions (due to limited optical visibility at high volume fractions) and tracer particles will influence coalescence and breakup of the dispersed phase. Numerical methods are a viable approach to study dispersed two-phase flows, thanks to the ongoing progress in the development of parallel computer systems and numerical algorithms. All nu-

merical methods for turbulent dispersed two-phase flows rely on the solution of a set of partial differential equations (PDE) to describe the motion of the phases. In these methods difficulties arise from the deformability of interfaces, discontinuities across interfaces, physical transfer processes taking place across interfaces, and the wide range of length and time scales involved. First, the deformable interface is part of the solution, and needs to be resolved on a fine mesh to accurately follow the interface motion. Second, physical properties (like pressure, viscosity, and density) show a sharp jump across the interface, which is usually hard to capture numerically. Next, across the interface between phases, transfer of momentum, heat, mass, and phase change can arise, which all need additional modeling. Finally, to fully resolve all details of the flow, all length and time scales should be resolved. Fig. 1.5

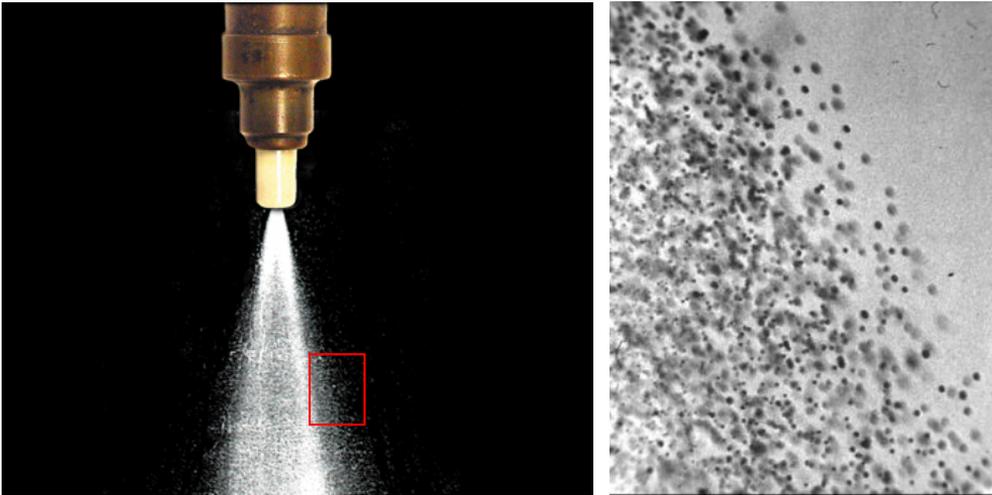


Figure 1.5: A diesel fuel spray (left) with a close-up of the diesel droplets at the edge of the spray (right). This figure depicts the large range of length scales between the diesel spray and its constituent droplets. Image sources: mdhmotors.com and icel.tkk.fi/research/Common_Rail.html.

shows the big range of length scales in a diesel fuel spray and its constituent droplets. The smallest flow scale in a turbulent flow is the Kolmogorov scale, η , where viscous effects become dominant and the energy is dissipated. The largest, energy containing flow scale, \mathcal{L} , is of the order of the length scale of the flow configuration that generates the turbulence. The ratio of the largest to smallest scales can be used as an indication for the total amount of computational work required. Kolmogorov [74] showed that

$$\mathcal{L}/\eta \sim Re^{3/4}, \quad (1.9)$$

which makes the total amount of grid cells for a fully resolved three-dimensional problem $\sim Re^{9/4}$. Together with the total number of computational time steps, which scales with $\sim Re^{1/2}$, the total number of numerical operations scales with $\sim Re^{11/4}$.

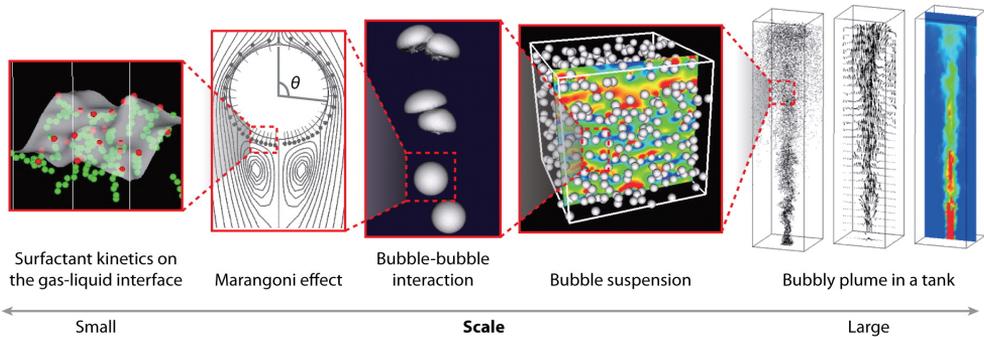


Figure 1.6: Multiscale structure of dispersed flows. Taken from Takagi and Matsumoto [148].

Fig. 1.6 shows the multiscale structure of dispersed flows, where the flow scales related to the interface are expected to be even much smaller than the Kolmogorov scale of the carrier phase. This implies that the number of operations in a fully resolved simulation up to the flow scales of the interface is even higher than the value given above. These difficulties limit the applicability of numerical methods that resolve all details of a turbulent dispersed two-phase flow.

Since not all flow problems require the solution of all the present scales, less expensive numerical methods have been developed which still provide adequate results for the problem under consideration. A common classification for turbulent dispersed two-phase flows is therefore given by the amount of detail they provide: Euler-Euler methods, Euler-Lagrange methods and Direct Numerical Simulation (DNS) of interfacial flows. For the design of large scale industrial applications Euler-Euler methods are usually a good choice. In these methods a continuum description for both phases is employed, by applying volume-averaging to the continuity and momentum equations. Each phase has its own continuity and momentum equations, which are coupled by the void fraction and interfacial momentum terms. This method incorporates two-way coupling by definition, and is especially suitable for flows with high dispersed phase volume fraction. A frequently used Euler-Euler method is the Two-Fluid Method (TFM). A more advanced group of methods to simulate dispersed two-phase flows are Euler-Lagrange methods. These methods treat the continuous phase in an Eulerian manner, and the dispersed phase in a Lagrangian manner. The continuous phase is described as a continuum, which is governed by single-phase flow equations. Each dispersed fluid particle is described by its own equation of motion. This leads to a fully decoupled system of equations, which are coupled by relations to model mass, momentum, and energy transfer between the phases. Due to their increased detail, Euler-Lagrange methods are generally more expensive than Euler-Euler methods. However, next to the design of industrial applications they can as well be used as a research tool. Frequently used Euler-Lagrange methods are the Discrete Particle Model (DPM) and the Discrete Bubble Model (DBM). The most accurate method in above given classification is

DNS of interfacial flows, where the two immiscible phases are each treated separately as a fluid. In DNS the governing equations, the so-called Navier-Stokes equations, are solved for each fluid on a sufficiently fine computational grid to ensure that all turbulent flow details are resolved. The interface between the two fluids is part of the solution of the system. The two phases are mathematically coupled by interface conditions, which express continuity of velocity and tangential stress. However, due to its expensive nature, the applicability of the method is limited to research problems.

Compared to DNS of interfacial flow, the main advantage of the Euler-Euler and Euler-Lagrange methods is their reduced computational cost. However, due to the averaging of the original flow equations, small-scale motions (e.g. viscous dissipation, individual bubble motion, interactions between bubbles, interaction between phases) are not resolved. These effects must be accounted for by so-called closure relations, which rely on the assumption that the unresolved small-scale motions can be related to resolved-scale motions in an universal and fairly simple manner. The quality (reliability) of these closure relations has a strong influence on the quality and applicability of the obtained results. Therefore, although DNS is usually inefficient for practical applications, it can provide physical insight and data for the creation of high quality closure relations. Thanks to the increasing computational power of supercomputers, more complex dispersed two-phase DNS have recently become in reach. The aim of the research presented in this thesis is to develop and validate an efficient numerical tool that allows DNS of many dispersed ($O(10^3)$) fluid particles in a turbulent carrier liquid. Furthermore, since coalescence and breakup events are likely to occur in such flows, models for these phenomena will also be developed and validated. The numerical tool is based on the work of Coyajee [26], who developed a method that was able to accurately simulate the buoyant rise of a limited number (8) of droplets in a periodic box. In this PhD research it will be further developed to improve fundamental understanding of two-phase flows, as well as to enable future validation and development of closure relations.

1.3.1 *Governing equations for interfacial two-phase flows*

In Direct Numerical Simulation (DNS) of interfacial two-phase flows the Navier-Stokes (NS) equations are solved in both the dispersed and the continuous phase. The interface between the phases is modeled as a moving, deformable interface, and is directly coupled with the fluid motion by appropriate interface conditions for pressure, viscosity, and density. This method allows detailed investigation of the interactions between the two phases, and the effect of dispersed fluid particles on each other. DNS results can therefore provide physical insight and closure relations for less detailed numerical methods. This section describes the fluid equations for both phases, interface conditions between the phases, and the mathematical interface description.

Fluid equations

If both fluid phases are considered to be isothermal, incompressible Newtonian fluids, and mass transfer is neglected, the fluid motion can be described by

$$\nabla \cdot \mathbf{u} = 0, \quad (1.10)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \boldsymbol{\tau} + \mathbf{f}_b + \mathbf{f}_{st}, \quad (1.11)$$

where $\mathbf{u} = (u, v, w)^T$ denotes the velocity vector, p the pressure, \mathbf{f}_b any body forces such as gravity, \mathbf{f}_{st} the surface tension force, and $\boldsymbol{\tau} = \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$ the viscous stress tensor. Eq. (1.10) describes conservation of mass, and Eq. (1.11) describes conservation of momentum. The latter are also referred to as the Navier-Stokes equations. The density, ρ , and viscosity, μ , are assumed constant within each phase and defined by ρ_0, μ_0 in fluid phase 'o' (continuous phase) and by ρ_1, μ_1 in fluid phase '1' (dispersed phase). If a phase indicator function χ is introduced as

$$\chi = \begin{cases} 0 & \text{in fluid phase 'o'} \\ 1 & \text{in fluid phase '1'} \end{cases}, \quad (1.12)$$

then ρ and μ can be expressed as

$$\rho = \rho_0 (1 - \chi) + \rho_1 \chi, \quad (1.13a)$$

$$\mu = \mu_0 (1 - \chi) + \mu_1 \chi. \quad (1.13b)$$

Interface conditions

Since the fluids are considered immiscible, their mutual interface Γ is a material property of the flow and its motion can be described by

$$\frac{\partial \Gamma}{\partial t} + \mathbf{u} \cdot \nabla \Gamma = 0, \quad (1.14)$$

where \mathbf{u} is the velocity of the fluid at the location of the interface. At this interface, the following kinematic and dynamic condition hold

$$[\mathbf{u}]_\Gamma = \mathbf{0}, \quad (1.15a)$$

$$[(-p\mathbf{I} + \boldsymbol{\tau}) \cdot \mathbf{n}]_\Gamma = -\sigma \kappa_\Gamma \mathbf{n}, \quad (1.15b)$$

where $[\cdot]_\Gamma$ denotes a jump across the interface Γ (from phase 'o' to fluid phase '1'), \mathbf{I} the unit tensor, \mathbf{n} the interface normal vector (which points from fluid phase 'o' to fluid phase '1'), σ the surface tension coefficient between the phases (assumed constant), and $\kappa_\Gamma \equiv -\nabla \cdot \mathbf{n}$ the interface curvature. Note that by this definition of the interface normal vector $\kappa_\Gamma > 0$, which is in accordance with the definition that a sphere has positive curvature. If the directions of the interface normal vector and the jump across the interface are reversed, (1.15) remains valid. Interface condition (1.15b) shows that the jump conditions for the pressure and velocity are coupled.

Kang et al. [69] derived that when the viscosity is modeled as being continuous across the interface, the derivatives of the velocity are also continuous, and (1.15) reduces to

$$[\nabla \mathbf{u}]_{\Gamma} = \mathbf{0} \tag{1.16a}$$

$$[p]_{\Gamma} = \sigma \kappa_{\Gamma}. \tag{1.16b}$$

Therefore, by using the assumption of a continuously varying viscosity across the interface, the jump conditions for the pressure and velocity are decoupled. There is no jump in the gradient of the velocity, which makes that derivatives of the velocity can be discretized by the same scheme in the entire computational domain. The only resulting interface condition is a jump in the pressure at the location of the interface, due to the surface tension force. Since the surface tension force is explicitly incorporated in the DNS method, no modeling is required for forces between the phases (which is usually split up into several parts, including the steady state drag and lift, added mass, Basset force) [155]. Results can therefore be used to verify existing and develop new relations for these terms. Since all fluid and contact dynamics is fully resolved, momentum coupling is also automatically accounted for.

Interface description

A first categorization of methods to resolve the interface is based on how the mesh and interface are connected, see Fig. 1.7. Gridded Interface Methods (GIM) are ideal

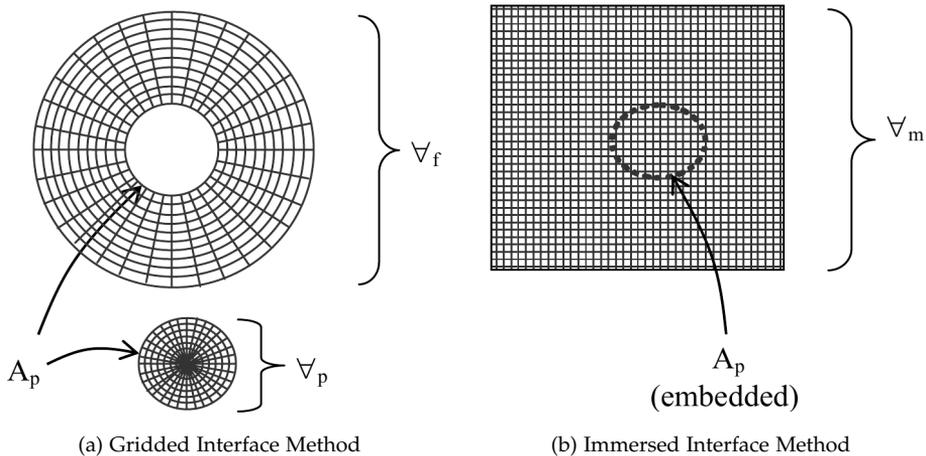


Figure 1.7: Two common methods used in interfacial flows. Taken from Loth [86].

for simple geometries which do not deform, and acquire a very accurate surface stress. The mesh is connected to the shape of the object, and therefore requires remeshing whenever the object moves or deforms. Immersed Interface Methods

(IIM) are ideal for complex, deformable geometries. The interface is immersed within a structured Eulerian mesh, and has a finite thickness. Since in this work deformable fluid particles are considered, only IIM will be discussed in more detail.

During the last decades several methods have been developed to accurately resolve the interface [127]. The best known methods are Front-Tracking (FT), Level-Set (LS), Volume-of-Fluid (VOF), see Fig. 1.8. The FT method explicitly tracks the

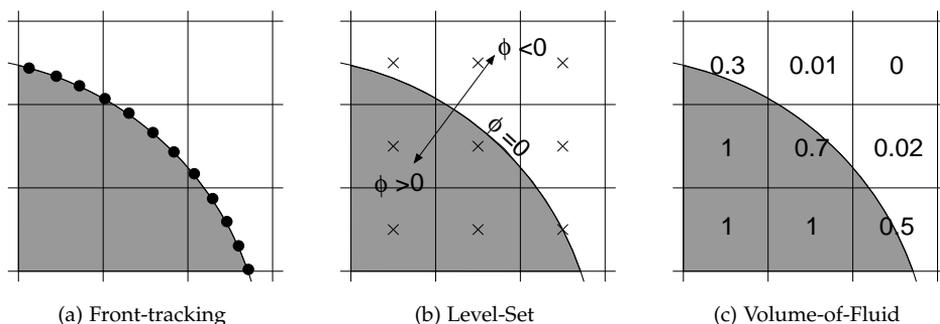


Figure 1.8: Commonly used immersed interface methods. Taken from Coyajee and Boersma [27].

interface during its motion through the continuous phase by marker particles in a Lagrangian manner. The LS and VOF methods implicitly capture the interface by a phase indicator function in an Eulerian manner, where the interface location is defined where this function changes property. Since each method has its own strengths and weaknesses, hybrid methods have been developed in an attempt to overcome the shortcomings of each individual method. Examples of such hybrid methods are the Hybrid Particle Level-Set (HPLS) method, the Level Contour Reconstruction (LCR) method, the mixed markers and VOF method, the Mass-Conserving Level-Set (MCLS) method, and the Coupled Level-Set Volume-of-Fluid (CLSVOF) method. These methods are explained in a more detail below.

The FT method was developed by Tryggvason and co-workers [107–109, 153], and represents the interface by a set of connected marker particles. These marker particles are advected with the local velocity, which is interpolated from the fixed grid to the marker particle locations. After advection of the marker particles, the interface is restructured to ensure a proper resolution and mass conservation. During this restructuring, marker particles can be moved, added, or removed. The surface tension forces can be computed by explicit polynomial fitting of a continuous surface through the marker particles [153], or directly from the triangular markers which represent the interface [32, 35, 132]. Next, these forces are conservatively interpolated by a weighting function to the Eulerian grid where the Navier-Stokes equations are solved. Advantages of the FT method are the accurate and straightforward representation of the interface. A disadvantage is that topological changes are less straightforward, and need special attention. The frequently used LS method [1, 21, 102, 112–114, 129, 130, 135, 142, 144–146], represents the interface by a signed

distance function, where the zero-level indicates the location of the interface. This function is advected by the local velocity field, which will however cause the LS to deviate from a signed distance function. Therefore, frequent reinitialization of the LS function is required to conserve it as a signed distance function. Advantages are the simple extraction of interface location, interfacial normal vectors, and curvature. Topological changes are automatically dealt with, which is generally considered as an advantage. However, these topological changes are only numerical phenomena and therefore not based on physics. An important disadvantage is the lack of conservation of mass. Although the LS function itself can be conserved, conservation of the enclosed volume is not ensured. The VOF method Hirt and Nichols [59], Kleefsman [72], Lörstad et al. [96], Noh and Woodward [110], Rudman [125], Youngs [165] defines different fluid phases on an Eulerian grid by the volume fraction of each fluid phase in a computational cell. Advantages are an excellent volume conservation, and an automatic treatment of topological changes. Disadvantages are the difficult interface reconstruction (a single volume fraction does not allow for a unique interface reconstruction), and its complex implementation in three dimensions. The HPLS method [39, 47, 48] places a set of Lagrangian marker particles randomly near the interface, which is defined by the zero LS. If a marker particle moves across the interface, an error in the LS representation of the interface has occurred. These errors are corrected by adjusting the advected LS, which thereby provides subgrid scale accuracy. Wang et al. [163] noted that the accuracy is sensitive to particle reseeding strategies and that the interface represented by the corrected LS set function is not smooth. Therefore, an improved particle correction procedure for the original HPLS method was developed, which was shown to outperform the LS and original HPLS method. It was shown that for more realistic problems the CLSVOF method slightly outperformed the HPLS method [162]. The LCR method [132, 133] circumvents the disadvantageous book-keeping of logical connections between neighboring marker particles of the original FT method. Furthermore, the interface is reconstructed by linear approximation of a contour of an indicator function. This reconstructed interface provides new marker particles, which periodically replace the original ones. Since the indicator function automatically handles topological changes, newly constructed interfaces also take on these changes. The mixed marker VOF method describes the interface using both an Eulerian VOF function and Lagrangian marker particles. The proposed method of Aulisa et al. [5] is only for two-dimensional problems, and the advection of the marker particles by a fourth-order Runge-Kutta method is the most expensive part of the algorithm. Nevertheless, the method allows tracking of filamentary interface structures thinner than the cell size [94]. The MCLS method [117] derives a VOF function from the LS function through analytical relations. Next, both the LS and derived VOF functions are advected to the next time level. The advected LS function is reinitialized by solving the Eikonal equation as an initial value problem, which preserves the LS function as a signed distance function. Possible truncations of the VOF function are corrected by redistributing the mass to neighboring cells. Finally, since the enclosed volume of the LS function can change, the advected VOF function is used to correct the LS function. The formal accuracy

of the MCLS method is only first order, due to first order discretization of the fluxes in the LS advection and the LS correction procedure Coyajee [26]. The CLSVOF method advects individual LS and VOF functions, where the LS function is used to describe the interface geometry and a separate VOF function to ensure volume conservation. After every time step, the LS and VOF functions are coupled through the interface normal vector (determined from the LS function), and the reconstructed interface (obtained from the VOF function). The formal accuracy of the CLSVOF method is second order [26], and therefore preferred above the MCLS method. In this work the multiple marker CLSVOF method [27] is employed, where each interface is described by its own LS and VOF function. The method and its efficiency improvements are explained in more detail in Chapter 2.

1.3.2 Dimensionless parameters

Fluid motion is caused by forces which act on fluid elements, which can be expressed by a characteristic length scale \mathcal{L} , velocity scale \mathcal{U} , continuous phase density ρ_c , continuous phase viscosity μ_c , gravitational acceleration g , and surface tension σ . These forces can be classified in three groups: volume/body forces, surface forces, and line forces. Volume forces, like the gravity and Archimedes force, act on a fluid element with volume $\mathcal{V} \propto \mathcal{L}^3$. The gravity force, $\mathbf{f}_g = \mathcal{V}g\rho_d$, tends to accelerate the fluid element in the direction of the gravity vector. The Archimedes force, $\mathbf{f}_A = \mathcal{V}g\rho_c$, is an opposite force equal to the weight of the displaced volume. The buoyancy force, $\mathbf{f}_b = \mathcal{V}g\Delta\rho_c$, is the difference between the gravity and Archimedes force. Surface forces, like pressure and viscous forces, act on a surface area $\mathcal{A} \propto \mathcal{L}^2$ of a fluid element. The pressure force, $\mathbf{f}_p = \mathcal{A}\Delta p$, tends to accelerate the fluid element in the direction opposite to the pressure gradient. The viscous force, $\mathbf{f}_v = \mathcal{A}\mu_c\mathcal{U}\mathcal{L}^{-1}$, tends to decrease local velocity differences due to viscosity. Line forces, like the surface tension force, act on a curve $\mathcal{S} \propto \mathcal{L}$ of the fluid element. The surface tension force, $\mathbf{f}_s = \mathcal{L}\sigma$, which is present between fluids, tends to decrease the surface area of the interface. This force is counteracted by a pressure difference across the interface.

All these forces are present in multiphase flow systems, but some are more important than others. Before a problem can be analyzed, insight is required in their importance with respect to each other. By forming dimensionless parameters, which express the ratio of two forces, the problem can be scaled. Problems with the same dimensionless parameters and the same boundary conditions lead to the same equations, and therefore will have the same solution. This is a useful advantage of scaling, since it enables comparison of different flow conditions.

In dispersed two-phase flows two important dimensionless parameters are the dispersed to continuous density and viscosity ratios: $\zeta = \rho_d/\rho_c$ and $\lambda = \mu_d/\mu_c$, respectively. Furthermore, from the forces described above, the following dimensionless parameters can be formed: Reynolds number, Froude number, Weber num-

ber, Eötvös number, Capillary number, and Morton number. These are respectively defined by

$$Re = \frac{\rho_c \mathcal{U} \mathcal{L}}{\mu_c} \quad (1.17)$$

$$Fr = \frac{\mathcal{U}^2}{g \mathcal{L}} \quad (1.18)$$

$$We = \frac{\rho_c \mathcal{U}^2 \mathcal{L}}{\sigma} \quad (1.19)$$

$$Eo = \frac{\Delta \rho g \mathcal{L}^2}{\sigma} = \frac{\Delta \rho}{\rho_c} \frac{We}{Fr} \quad (1.20)$$

$$Ca = \frac{\mu_c \mathcal{U}}{\sigma} = \frac{We}{Re} \quad (1.21)$$

$$Mo = \frac{g \Delta \rho \mu_c^4}{\rho_c^2 \sigma^3} = \frac{Eo We^2}{Re^4} = \frac{\Delta \rho}{\rho_c} \frac{We^3}{Fr Re^4} \quad (1.22)$$

The Morton number depends on fluid properties, but is independent of length and velocity scales, and therefore constant in an incompressible isothermal two-phase system. To illustrate the usefulness of dimensionless parameters, Fig. 1.9 shows the correlation of just three dimensionless parameters to show the shape regimes for fluid particles rising or settling in an unbounded domain.

The system (1.10)-(1.11) can be scaled using characteristic length scale \mathcal{L} , velocity scale \mathcal{U} , time scale $\mathcal{T} = \mathcal{L}/\mathcal{U}$, density ρ_c , and viscosity μ_c . The following equations are then obtained, where the prime* indicates a scaled variable

$$\nabla \cdot \mathbf{u}' = 0, \quad (1.23)$$

$$\frac{\partial \mathbf{u}'}{\partial t'} + \mathbf{u}' \cdot \nabla \mathbf{u}' = -\frac{1}{\rho'} \nabla p' + \frac{1}{\rho' Re} \nabla \cdot \boldsymbol{\tau}' + \frac{1}{Fr} \hat{\mathbf{g}}. \quad (1.24)$$

Here $\boldsymbol{\tau}' = \mu' (\nabla \mathbf{u}' + \nabla \mathbf{u}'^T)$ denotes the dimensionless viscous stress tensor, and $\hat{\mathbf{g}}$ is a unit vector that points in the direction of the gravitational acceleration. The system of dynamic conditions at the interface (1.15a)-(1.15b) reads in dimensionless form

$$[\mathbf{u}']_{\Gamma} = 0, \quad (1.25)$$

$$\left[\left(-p' \mathbf{I} + \frac{1}{Re} \boldsymbol{\tau}' \right) \cdot \mathbf{n} \right]_{\Gamma} = -\frac{1}{We} \kappa_{\Gamma} \mathbf{n}, \quad (1.26)$$

The dimensionless density and viscosity are given by

$$\rho' = 1 + (\zeta - 1)H(\phi) \quad (1.27)$$

$$\mu' = 1 + (\lambda - 1)H(\phi). \quad (1.28)$$

1.4 RESEARCH OBJECTIVES

Since many industrial and environmental processes can be considered as a turbulent dispersed two-phase flows, detailed investigation of such flows is of general interest.

* For clarity, in the remainder this prime is always dropped.

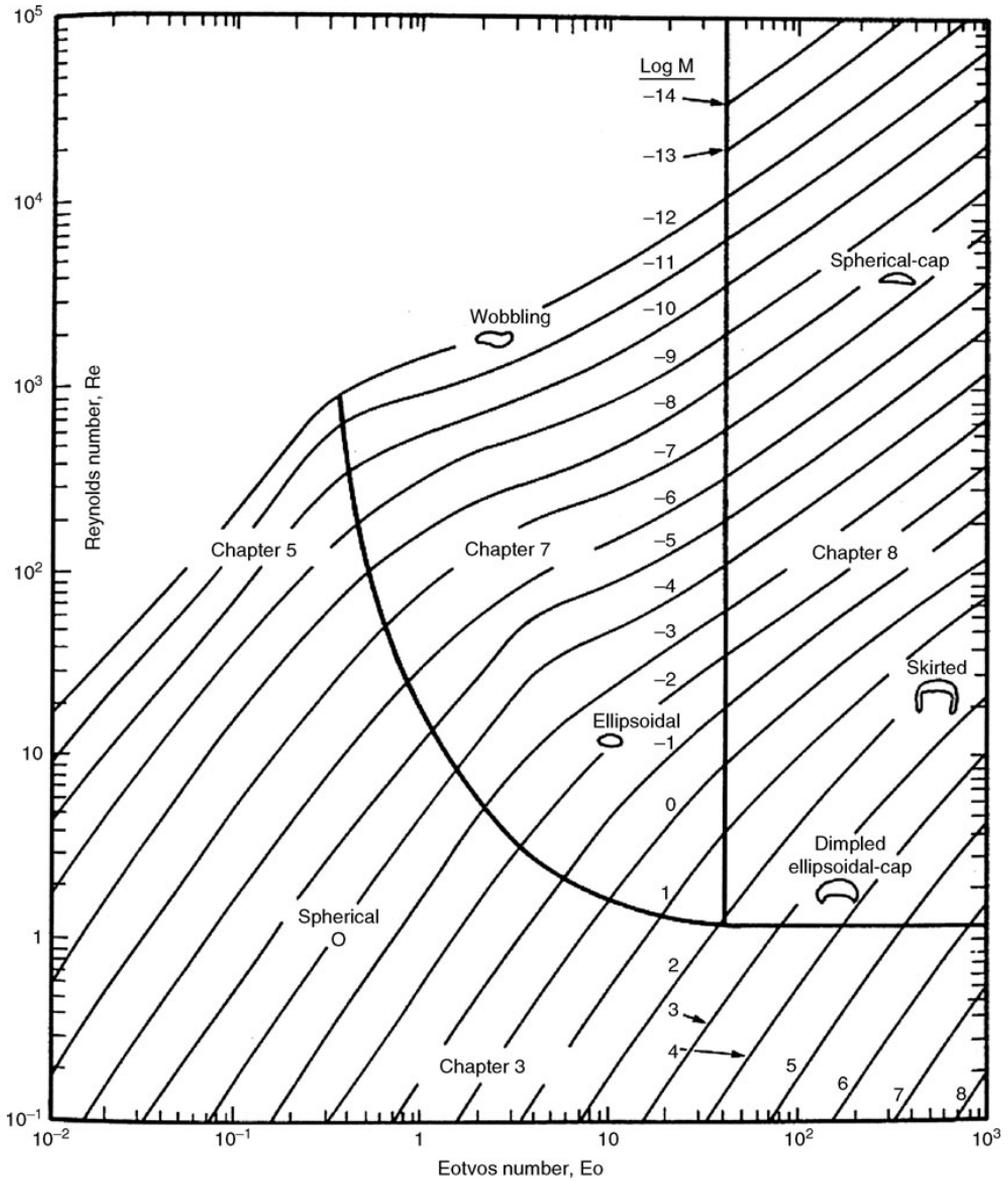


Figure 1.9: Fluid particle shape regimes as a function of Reynolds and Eötvös number with the Morton number as a parameter. This graph is a reproduction from Clift et al. [24], based on the original publication by Grace et al. [52].

Although free surface flow simulations were already performed about half a century ago by Harlow and Welch [54], solving the full three-dimensional Navier-Stokes equations with large mass density and viscosity jumps combined with deformable interfaces remains to be an elaborative task. Besides the difficulties of interface jump conditions and mass-conserving interface advection, in dense dispersed flows coalescence and breakup events also become important. The in-house code of the multiple marker CLSVOF method of Coyajee [26] serves as a starting point for this research. By application of separate markers functions for each interface, the buoyant rise of 8 deformable fluid particles without so-called numerical coalescence could be simulated with this code [27]. Therefore, the overall goal of this work is to investigate the feasibility of the CLSVOF method with extensions for coalescence and breakup to simulate turbulent dispersed two-phase flows. To reach this goal, the research can be subdivided in three milestones:

1. **Efficiency improvements**

One of the goals of this work is to simulate dense dispersed flows, which implies that the method has to be improved so that it can handle $\mathcal{O}(10^3)$ dispersed fluid particles. To allow for such large scale computations, the method has to scale up well with increasing number of fluid particles. Since Coyajee [26] noted that over 80% of the entire computational effort was required to solve the pressure Poisson equation, efficiency improvement of this solver will be of key interest. Furthermore, since the code will be run on a distributed system, the parallelization of both the continuous and dispersed phases will be optimized for this setup.

2. **Coalescence/breakup modeling**

As the original in-house code lacks physics-based coalescence and breakup, these physical phenomena have to be explicitly added and validated against available experimental and/or numerical results. Furthermore, the computational overhead of these additional physics should remain limited to allow large scale simulations.

3. **Large scale simulations**

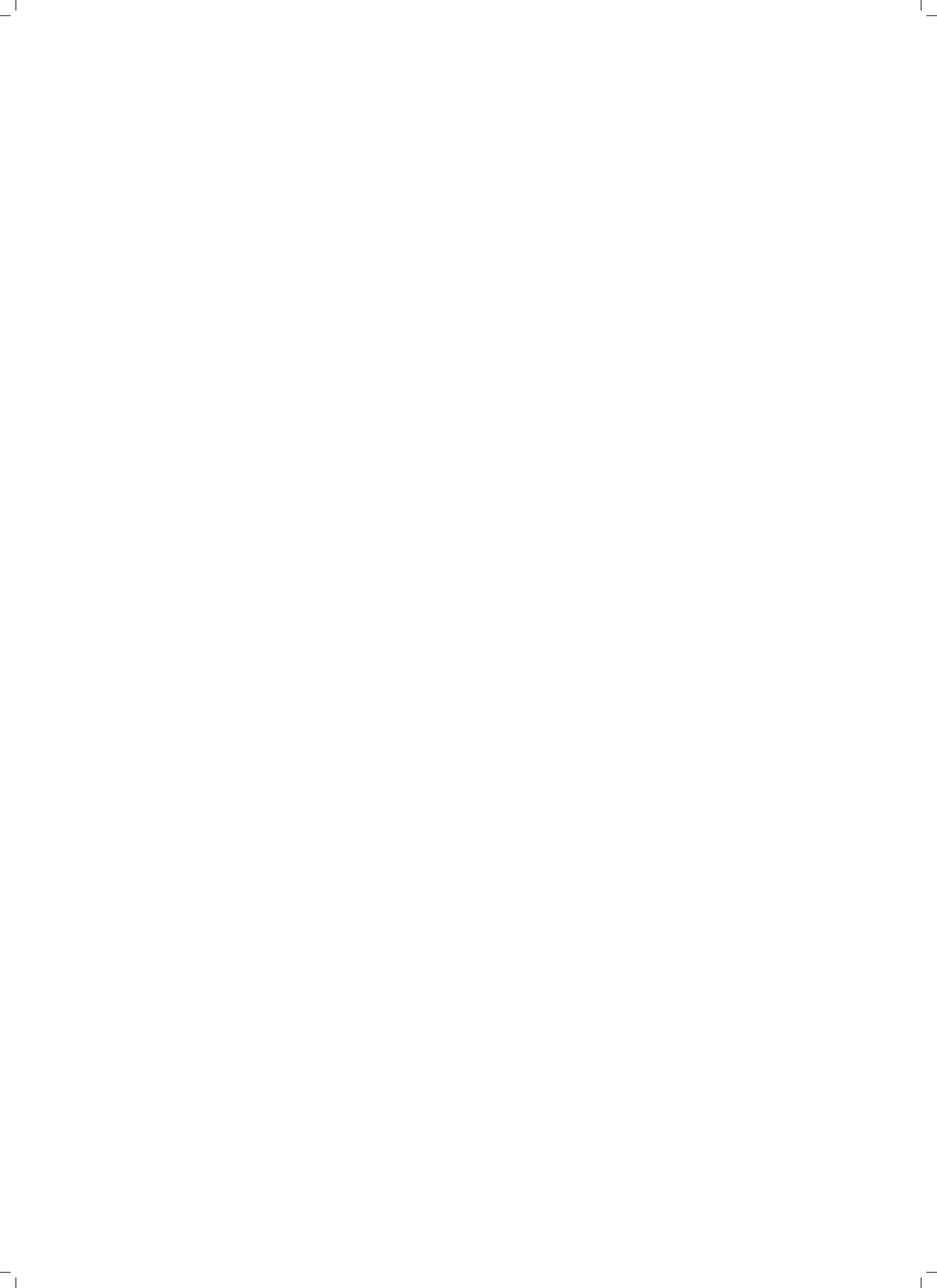
If the efficiency improvements are sufficient, large scale simulations of a turbulent dispersed two-phase flow (without coalescence/breakup modeling) can be done. These results shall be validated against existing available experimental and/or numerical results. Next, when the coalescence/breakup model is fully functional, the effect of coalescence/breakup of many fluid particles in a turbulent flow can be quantified.

1.5 THESIS OUTLINE

The present thesis describes the development of a numerical tool for the simulation of many dispersed fluid particles in a turbulent carrier fluid. This numerical tool is coupled with coalescence and breakup modules, to allow an integrated simulation

of dispersed fluid particles with topological changes. The outline of this thesis is as follows.

Chapter 2 describes the development of the numerical tool to simulate fully resolved dispersed two-phase flows. To be able to simulate many dispersed fluid particles within reasonable time, several efficiency improvements for the in-house code are presented. In Chapter 3, the modeling of coalescence and breakup is described. The developed modules for both phenomena are presented and validated against available experimental work. To investigate the feasibility to simulate turbulent dispersed two-phase flows with many fluid particles, simulations without the coalescence/breakup extension modules have been performed. These preliminary results are presented in Chapter 4. Due to difficulties encountered, which are described in detail, these simulations conclude this research. Finally, Chapter 5 summarizes the results and conclusions of the present work in its whole. Furthermore, recommendations for future work in turbulent dispersed two-phase flow research are given.



Efficiency is intelligent laziness

— David Dunham

Abstract *The efficiency of the CLSVOF method of Coyajee and Boersma [27] is improved to accommodate the simulation of a large number, $\mathcal{O}(10^2 - 10^3)$, of droplets dispersed in a turbulent carrier fluid. Firstly, the implementation of a local multiple marker method removes the strong increase in computational time with the number of droplets. Secondly, the deflation method [150] is used as a second-level preconditioner in the pressure solver to reduce the number of iterations. Finally, three-dimensional domain decomposition is used instead of a one-dimensional one. For the buoyant rise of 48 droplets in a periodic box an overall speed-up by a factor eight is achieved.*

2.1 INTRODUCTION

Multiphase flows play an important role in many geophysical and industrial processes including formation of droplets in clouds, separation in oil production, and emulsification in food industry. Due to their complexity, however, multiphase flows are by far less understood than single-phase flows. By lack of computational power and efficient numerical methods in the past, multiphase flows have therefore received less attention in Computational Fluid Dynamics (CFD) than single-phase flows. Nevertheless, still many articles on CFD for multiphase flows have been published over the last five decades [44, 54, 56, 59, 114, 127, 157].

Of key interest are in particular dense turbulent two-phase flows, where a large number of fluid particles are dispersed in a turbulent carrier fluid. Examples of such flows are droplet formation in clouds [131], fuel injection in engines [99, 134, 137] and bubble column reactors [98]. For these flows the main questions are how droplets move within the flow, interact with each other, and modulate the behavior of the turbulent carrier fluid. To be able to simulate such complex flow physics, a very accurate and efficient numerical method is needed. This method must also be robust and scalable with respect to among others the number of droplets, the grid resolution and density ratio.

Direct Numerical Simulations (DNS) proved to be very accurate and form a perfect tool for studying the structure and dynamics of turbulence [100]. To be able to simulate moving and deformable interfaces, the most commonly used methods are

† This chapter has been published in *Computers & Fluids* [75]

front-capturing and front-tracking methods. A front-capturing method defines interfaces implicitly by a marker function on a fixed grid. One can use a Level-Set (LS) [144], Volume-of-Fluid (VOF) [125] or a combined LS/VOF function [143] to capture the dynamics of the interface. A disadvantage of the implicit definition of the interfaces is the occurrence of artificial numerical coalescence when two interfaces are close to each other. Coyajee and Boersma [27] used separate marker functions for each interface to overcome this problem. A front-tracking method [157] defines the interface explicitly by a set of logically connected particles on a fixed grid. The dynamics of the interfaces are followed by tracking these particles in a Lagrangian way. A disadvantage of the explicit definition of the interfaces is the need for re-meshing of the advected particles to maintain a consistent interface description. Also topological changes are not as straightforward as with the front-capturing method, which can deal with these changes automatically. However, in both methods the thin film lubrication and coalescence process is a subgrid phenomenon which might require additional modelling. Therefore, it is difficult to accommodate coalescence of two interfaces with physical realism without such a model.

Since we are interested in dense droplet-laden turbulent flows, coalescence and break-up will be of great importance. The chosen method is therefore the Coupled Level-Set Volume-of-Fluid (CLSVOF) method of Sussman and Puckett [143] with the multiple marker method of Coyajee and Boersma [27]. By combining this method with DNS and a physically realistic coalescence and breakup model, the method will yield very detailed information about the dynamics of a swarm of droplets in a turbulent carrier fluid. It is therefore suitable for academic studies with the aim to improve understanding of droplet dynamics and turbulence modification. Due to the high number of droplets, it is necessary to have a very efficient parallel code with a low storage complexity.

In this work an improved version of the method developed by Coyajee and Boersma [27] is presented, which is able to simulate systems with a large number of droplets accurately within reasonable time. Since Coyajee and Boersma [27] defined each droplet in the entire three-dimensional domain, which requires a large amount of memory storage and computational effort, here a localized version of their multiple marker method is used. Coyajee [26] reported that more than 80% of the total computational time was spent in solving the variable coefficient Poisson equation with the preconditioned Conjugate Gradient (PCG) method.

Following Tang [150], the deflation method is applied to PCG, which leads to a better conditioned system and therefore a more efficient solver. Finally, the parallelization of the method is modified, to improve the load balancing and decrease the amount of communication.

In Section 2.2 the mathematical model used to describe the motion of the fluid and appropriate interface conditions are formulated. This section also contains the discretization of these equations and some numerical implementation issues. The proposed improvements and their implementations are explained in Section 2.3. The influence of these improvements on the efficiency of the code are presented in Section 2.4. Finally conclusions and ideas for future research are given in Section 2.5.

2.2 MULTIPLE MARKER FRONT-CAPTURING METHOD

The motion of two immiscible fluid phases, where one fluid phase is dispersed in a surrounding continuous fluid phase, can be described by combining the equations for the fluid motion with some appropriate interface conditions. The motion of the interface Γ depends on the velocity field of the fluid at the location of the interface, which makes this a moving boundary problem. By demanding continuity of velocity and tangential stress at the interface, both fluid phases are coupled. Density, ρ , and viscosity, μ , are discontinuous across the interface. The surface tension forces are local forces, which occur only at the interface location. The moving boundary problem can be simplified by regularizing the viscosity, which reduces the discontinuity in normal stress at the interface to a pressure jump only.

2.2.1 Governing equations

If both fluid phases are considered to be isothermal incompressible Newtonian fluids, the fluid motion can be described by

$$\nabla \cdot \mathbf{u} = 0, \quad (2.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \boldsymbol{\tau} + \mathbf{f}_b + \mathbf{f}_{st}, \quad (2.2)$$

where $\mathbf{u} = (u, v, w)^T$ denotes the velocity vector, p the pressure, \mathbf{f}_b any body forces such as gravity, \mathbf{f}_{st} the surface tension force, and $\boldsymbol{\tau} = \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$ the viscous stress tensor. The density, ρ , and viscosity, μ , are assumed constant within each phase and defined by ρ_0, μ_0 in fluid phase 'o' (continuous phase) and by ρ_1, μ_1 in fluid phase '1' (dispersed phase). If a phase indicator function χ is introduced as

$$\chi = \begin{cases} 0 & \text{in fluid phase 'o'} \\ 1 & \text{in fluid phase '1'}, \end{cases} \quad (2.3)$$

then ρ and μ can be expressed as

$$\rho = \rho_0 (1 - \chi) + \rho_1 \chi, \quad (2.4)$$

$$\mu = \mu_0 (1 - \chi) + \mu_1 \chi. \quad (2.5)$$

Since the fluids are considered immiscible, their mutual interface Γ is a material property of the flow and its motion can be described by

$$\frac{\partial \Gamma}{\partial t} + \mathbf{u} \cdot \nabla \Gamma = 0, \quad (2.6)$$

where \mathbf{u} is the velocity of the fluid at the location of the interface in the absence of mass transfer between the two phases.

At this interface, the following kinematic and dynamic conditions hold

$$[\mathbf{u}]_{\Gamma} = 0, \quad (2.7)$$

$$[(-p\mathbf{I} + \boldsymbol{\tau}) \cdot \mathbf{n}]_{\Gamma} = -\sigma \kappa_{\Gamma} \mathbf{n}, \quad (2.8)$$

where $[\cdot]_\Gamma$ denotes a jump across the interface Γ , \mathbf{I} the unit tensor, \mathbf{n} the interface normal vector (which points from fluid phase ‘0’ to fluid phase ‘1’), σ the surface tension coefficient between the phases (assumed constant), and $\kappa_\Gamma \equiv -\nabla \cdot \mathbf{n}$ the interface curvature. Interface condition Eq. (2.8) shows that the jump conditions for the pressure and velocity are coupled. Kang et al. [69] derived that when the viscosity is continuous across the interface, the derivatives of the velocity are also continuous, and Eq. (2.8) reduces to

$$[\nabla \mathbf{u}]_\Gamma = \mathbf{0} \quad (2.9)$$

$$[p]_\Gamma = \sigma \kappa_\Gamma. \quad (2.10)$$

Therefore, by using a continuous viscosity, the jump conditions for the pressure and velocity are decoupled. There is no jump in the gradient of the velocity, which makes that derivatives of the velocity can be discretized by the same scheme in the entire computational domain. The only resulting interface condition is a jump in the pressure at the location of the interface, due to the surface tension force.

2.2.2 Discretization of the Navier-Stokes equations

Since the fluid motion and the moving interfaces are coupled, for simplicity the system (2.1)-(2.2) and (2.6) are solved sequentially in time. Eqs. (2.1) and (2.2) are solved on a fixed, equidistant Cartesian grid in a three-dimensional rectangular domain using an incremental pressure-correction method [68]. A staggered arrangement of variables is used [54]: velocity components are located at cell faces, whereas pressure, density, viscosity and marker functions are defined at cell centers. The tentative velocity \mathbf{u}^* is computed by the PCG method from the predictor step

$$\begin{aligned} \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = & \\ & -\frac{3}{2}\mathcal{A}(\mathbf{u}^n) + \frac{1}{2}\mathcal{A}(\mathbf{u}^{n-1}) + \\ & \frac{1}{\rho^{n+\frac{1}{2}}} \left(\mathcal{B}_{\text{imp}}(\mathbf{u}^*) + \mathcal{B}_{\text{exp}}(\mathbf{u}^n) - \mathcal{G}p^{n-\frac{1}{2}} \right) + \mathbf{g}, \end{aligned} \quad (2.11)$$

where \mathbf{u}^n denotes the velocities at time level n , \mathcal{A} the discrete convective operator, \mathcal{B} the discrete diffusive operator and \mathcal{G} the discrete gradient operator. The convective term is integrated explicitly by the second order Adams-Bashforth (AB2) method. The diffusive term is split and integrated by the implicit Euler method (\mathcal{B}_{imp}) and the explicit Euler method (\mathcal{B}_{exp}). The pressure, $p^{n-\frac{1}{2}}$, is the pressure at the old time level $n - \frac{1}{2}$, and also includes the pressure jump as given by Eq. (2.10). For the exact discretization of this term see Section 2.2.4.

The time step Δt is determined from an adaptive criterion based on the restrictions due to convection [144], surface tension [9] and body forces [144]. In the numerical experiment considered in Section 2.4 the surface tension time step constraint is limiting. This constraint is given by

$$\Delta t_s = \frac{1}{2} \sqrt{\frac{We \max(\zeta, 1)}{8\pi}} h^{3/2}, \quad (2.12)$$

where We is the droplet Weber number, ζ the density ratio ρ_1/ρ_0 and h the uniform mesh width. The Weber number is defined as

$$We = \frac{\rho_0 g D^2}{\sigma} \quad (2.13)$$

where ρ_0 is the continuous phase density, g the gravitational acceleration, D the droplet diameter, and σ the surface tension coefficient.

The velocities at time level $n + 1$ are computed from the corrector step

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{\mathcal{G}p^*}{\rho^{n+\frac{1}{2}}}, \quad (2.14)$$

where

$$p^* = p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}. \quad (2.15)$$

As before, in Eq. (2.15) the surface tension is included in the pressure terms. By taking the divergence of Eq. (2.14) and under the constraint of Eq. (2.1) this yields

$$\mathcal{D} \left(\frac{1}{\rho^{n+\frac{1}{2}}} \mathcal{G}p^* \right) = \frac{1}{\Delta t} \mathcal{D}(\mathbf{u}^*), \quad (2.16)$$

where \mathcal{D} denotes the discrete divergence operator. More generally, the linear system Eq. (2.16) can be written as

$$A\mathbf{x} = \mathbf{b}, \quad A \in \mathbb{R}^{n \times n}, \quad n \in \mathbb{N}, \quad (2.17)$$

where A is a real, symmetric, and positive semi-definite (SPSD) matrix. It can be proved [7] that the matrix A has exactly one zero eigenvalue. A well-known method to solve this system is the Conjugate Gradient (CG) method, which is described in Section 2.2.6.

2.2.3 Interface representation

The Level-Set (LS) function, ϕ , is used as the phase indicator function χ , in order to capture the interface. The interface itself is represented by the zero level of this marker function, so $\phi = 0$. Away from the interface, ϕ is required to be a distance function to the interface such that $\phi < 0$ in phase 'o' and $\phi > 0$ in phase 'i'. Now the Heaviside function $H(\phi)$ can be defined as

$$H(\phi) = \begin{cases} 1 & \text{if } \phi > 0 \\ 0 & \text{otherwise,} \end{cases} \quad (2.18)$$

which is equivalent to the already defined χ of Eq. (2.3). As described in Section 2.2.1 and following [144] and [21], the viscosity is regularized by replacing $H(\phi)$ with a continuous Heaviside function $H_\alpha(\phi)$

$$H_\alpha(\phi) = \begin{cases} 0 & \text{if } \phi < -\alpha \\ \frac{1}{2} \left(1 + \frac{\phi}{\alpha} + \frac{1}{\pi} \sin \left(\frac{\pi\phi}{\alpha} \right) \right) & \text{if } |\phi| \leq \alpha \\ 1 & \text{if } \phi > \alpha. \end{cases} \quad (2.19)$$

The density at cell centers is directly obtained from Eq. (2.18). However, the discretization of Eq. (2.16) requires values of $\frac{1}{\rho}$ at cell faces. In the vicinity of the interface these values are determined from the weighted harmonic average [82]

$$\frac{1}{\rho_{i+\frac{1}{2}}} = \frac{1}{\rho_i\theta + \rho_{i+1}(1-\theta)}, \quad (2.20)$$

where θ is given by

$$\theta = \frac{|\phi_i|}{|\phi_i| + |\phi_{i+1}|}. \quad (2.21)$$

Since the fluid phases are immiscible, the interface is a material property and the motion of the interface is performed by advecting the LS function

$$\phi_t + \mathbf{u} \cdot \nabla \phi = 0. \quad (2.22)$$

The solution of Eq. (2.22) does not conserve the volume enclosed by the LS function. Therefore, long-term computations of dispersed flow would lead to significant mass losses. By combining the LS function with a Volume-of-Fluid (VOF) function, the volume conservation can be improved. The VOF function, ψ , represents the discrete volume fraction in a computational cell and is defined as

$$\psi(t) = \frac{1}{\Delta x \Delta y \Delta z} \int_{\Omega} H(\phi(\mathbf{x}, t)) d\mathbf{x}, \quad (2.23)$$

where Ω denotes the volume of a computational cell. If the entire cell belongs to phase 'o', then $\psi = 0$; if the entire cell belongs to phase '1', then $\psi = 1$; and if it is a mixed cell (crossed by the interface), then $0 < \psi < 1$. The VOF is advected by the following mass conservation equation for the volume fraction

$$\psi_t + \nabla \cdot (\mathbf{u}\psi) = 0. \quad (2.24)$$

Since the VOF does not give any information about the distribution of the fluid inside a cell, the LS function is used to determine the amount of fluid that flows across the boundaries of the cell. The method used in this work is known as the Coupled Level-Set Volume-of-Fluid (CLSVOF) method of Sussman and Puckett [143], which consists of the following steps. First, $\psi^{n-\frac{1}{2}}$ and $\phi^{n-\frac{1}{2}}$ are advected by the local velocity field \mathbf{u}^n to obtain $\psi^{n+\frac{1}{2}}$ and ϕ^* . Next, since ϕ^* does not conserve the enclosed volume, ϕ^* is reinitialized by $\psi^{n+\frac{1}{2}}$ to obtain $\phi^{n+\frac{1}{2}}$, which does conserve the volume. Finally, the $\phi^{n+\frac{1}{2}}$ values of interface cells are extrapolated to four cells on either side of the interface to keep $\phi^{n+\frac{1}{2}}$ a distance function.

2.2.4 Surface tension

The surface tension forces are explicitly incorporated in the discretization by the Ghost Fluid (GF) method of Liu et al. [82]. Interface condition Eq. (2.10) shows that the surface tension forces lead to a jump condition for the pressure at the location of the interface. The GF method corrects the pressure gradient over the interface by

taking this jump into account. For example, in x -direction and at location $i + \frac{1}{2}$ the pressure gradient would become

$$\left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2}} = \frac{p_{i+1} - p_i - [p]_\Gamma}{\Delta x}, \quad (2.25)$$

where $[p]_\Gamma = \sigma\kappa_\Gamma$ if $\phi_i \leq 0$ and $\phi_{i+1} > 0$; $[p]_\Gamma = -\sigma\kappa_\Gamma$ if $\phi_i > 0$ and $\phi_{i+1} \leq 0$; and $[p]_\Gamma = 0$ in all other cases.

The interface curvature κ_Γ can be determined from the LS function or the VOF function. However, Coyajee and Boersma [27] showed that due to the reinitialization of the LS function from the VOF function, the reconstructed LS function is only second order accurate. Since the curvature is the second derivative of the LS function [26], the curvature will not converge with grid refinement. By using the height function (HF) method [46, 55, 96, 141], the curvature can be determined with second order accuracy from the VOF function. Since the curvature is required at the location of the interface Γ , the curvature is interpolated from the grid points to the interface [46].

2.2.5 Parallel implementation

The numerical method is parallelized by a domain decomposition approach, which is based on a division of the domain into non-overlapping subdomains [87]. Each processing element (PE) of the parallel computer handles one subdomain, which all contain approximately the same amount of grid points. At each grid point the discrete variables of the fluid and interface motion are integrated in time. The amount of work per grid point is assumed to be constant, so for the fluid and interface motion static load balancing is achieved. To be able to evaluate finite differences in an entire subdomain, halo regions are added at the border of the subdomains. The grid cell values in these halo regions are updated from neighboring subdomains when necessary. Communication between the PEs (subdomains) is performed by the Message Passing Interface (MPI) library software. Coyajee [26] used a sliced (one-dimensional) division, but also noted that such a division could be suboptimal compared to a multidimensional division. Therefore, three-dimensional domain decomposition has been implemented, which is discussed in Section 2.3.3.

Furthermore, Coyajee and Boersma [27] used a global multiple marker method, where multiple LS and VOF functions are defined in the entire computational domain. Although the motion (advection) of the interfaces leads to a balanced load of the PEs, the reinitialization does not. Since the reinitialization is an expensive operation, it is performed only in a narrow band [1, 143] around the interface. Only PEs which hold part of this narrow band are occupied by the reinitialization procedure, which therefore leads to an unbalanced load between PEs. In Section 2.3.1 a localized multiple marker method is introduced, which also includes an improved static load balancing for the interface advection and reinitialization.

2.2.6 Conjugate Gradient method

The Conjugate Gradient (CG) method is a common used iterative method for solving the SPSD linear system (2.17). The convergence of this method depends highly on the condition number $\kappa(A)$, which is defined as the ratio of the largest and the smallest nonzero eigenvalues of A . Tang [150] showed that Eq. (2.16) becomes more ill-conditioned if the number of interfaces is larger, the grid resolution is increased, or the density ratio grows.

The conditioning of A can be improved by applying a preconditioner, which tries to cluster the eigenvalues around 1. An effective preconditioner improves the convergence of the iterative method, sufficiently to overcome the additional cost of the preconditioner. Instead of solving Eq. (2.17), the preconditioned linear system

$$M^{-1}Ax = M^{-1}b \quad (2.26)$$

is solved, where M^{-1} is a $n \times n$ SPD preconditioner. The preconditioned Conjugate Gradient (PCG) method is implemented as shown in Algorithm 1, where x_{it} is the final solution of the iterative process. Coyajee and Boersma [27] used a block

Algorithm 1 Preconditioned CG (PCG) algorithm

- 1: Select x_0 . Compute $r_0 = b - Ax_0$,
 - 2: solve $My_0 = r_0$ and set $p_0 := y_0$.
 - 3: **for** $j := 0, \dots$, until convergence **do**
 - 4: $w_j := Ap_j$
 - 5: $\alpha_j := \frac{(r_j, y_j)}{(p_j, w_j)}$
 - 6: $x_{j+1} := x_j + \alpha_j p_j$
 - 7: $r_{j+1} := r_j - \alpha_j w_j$
 - 8: Solve $My_{j+1} = r_{j+1}$
 - 9: $\beta_j := \frac{(r_{j+1}, y_{j+1})}{(r_j, y_j)}$
 - 10: $p_{j+1} := y_{j+1} + \beta_j p_j$
 - 11: **end for**
 - 12: $x_{it} := x_{j+1}$
-

incomplete Cholesky (BIC) decomposition without fill-in, which is based on a non-overlapping data distribution and efficient for parallel use [51].

As starting vector, x_0 , the zero vector is chosen, since the convergence rate of the iterative process hardly depends on it [150]. The iterative process is terminated if the following termination criterion is reached

$$\frac{\|r_{j+1}\|_2}{\|r_0\|_2} < \delta \quad (2.27)$$

or

$$\frac{\|y_{j+1}\|_2}{\|y_0\|_2} = \frac{\|M^{-1}r_{j+1}\|_2}{\|M^{-1}r_0\|_2} < \delta \quad (2.28)$$

where r_{j+1} and y_{j+1} represents the original and preconditioned residual at iteration $j + 1$, respectively. The tolerance, $\delta > 0$, determines the accuracy of the solution and is a user-supplied parameter. For a more detailed discussion see Tang [150]. Coyajee [26] reported that solving the variable Poisson equation Eq. (2.16) required more than 80% of the total CPU time. In Section 2.3.2 the deflated PCG method will be introduced to decrease this portion.

2.3 NUMERICAL IMPROVEMENTS

2.3.1 Local multiple marker method

Coyajee and Boersma [27] used a global definition of the marker functions, where the LS and VOF functions are updated sequentially in the entire computational domain. The disadvantage of a global definition is its high storage complexity, which for the multiple marker method scales linearly with the amount of interfaces. Furthermore, the sequential update of the interfaces leads to a poor load balancing. The wall clock time needed to update the interface also scales linearly with the amount of interfaces. By localizing the marker functions and using a master/slave technique both problems can be alleviated. Therefore, the local multiple marker method leads to a lower storage complexity and a better load balancing. To reduce the storage complexity several methods have been proposed: octree method [85, 140], sparse block grid method [13], run-length encoding (RLE) method [61, 104] and a point-based method [25]. Some of these methods have the disadvantage that the access time depends on the amount of stored data. To be able to simulate a large number of droplets, it is essential to have a low access time. Therefore, here a dense, local, and dynamic box around an interface is used, which has $\mathcal{O}(1)$ access times. This box is just large enough to include all the cells of the narrow band.

Algorithm 2 Original update subroutine

```

1: Given  $\mathbf{u}^n, \phi^{n-\frac{1}{2}}, \psi^{n-\frac{1}{2}}$ .
2: for all interfaces do
3:   Advect  $\psi^{n-\frac{1}{2}}$  with  $\mathbf{u}^n$  to  $\psi^{n+\frac{1}{2}}$ 
4:   Advect  $\phi^{n-\frac{1}{2}}$  with  $\mathbf{u}^n$  to  $\phi^*$ 
5: end for
6: for all interfaces do
7:   Reinitialize  $\phi^*$  to  $\phi^{n+\frac{1}{2}}$ 
8: end for

```

The interface update procedure as used in [27] is described in Algorithm 2. This algorithm shows that the interfaces are first sequentially advected and afterwards, also sequentially, reinitialized. During the advection and reinitialization multiple (blocking) synchronizations between PEs are necessary since interfaces can overlap multiple PEs. Due to the global definition of the marker functions, all PEs are involved in these synchronizations, even if the interface is not within their own

domain. This leads to a strong coupling between the communication and the computation during the advection of the LS and VOF and the reinitialization of the LS.

However, since interfaces are not mutually coupled, the advection and reinitialization can be combined in just one loop. This is possible if the entire update of an interface is done by one PE, which is called the master. Algorithm 3 shows the improved update procedure, which consists of the following three steps. First, the master receives the local velocity fields from the slaves, which hold the physical interface and therefore have the local velocity fields. Second, the master processes update all interfaces in parallel. Finally, the updated marker functions are synchronized between the master and its slaves, which need this information to determine the local density and viscosity values.

Algorithm 3 Improved update subroutine

```

1: for all interfaces do
2:   Synchronize  $\mathbf{u}^n$  from slaves to master
3: end for
4: for all interfaces do
5:   if master then
6:     Advect  $\psi^{n-\frac{1}{2}}$  with  $\mathbf{u}^n$  to  $\psi^{n+\frac{1}{2}}$ 
7:     Advect  $\phi^{n-\frac{1}{2}}$  with  $\mathbf{u}^n$  to  $\phi^*$ 
8:     Reinitialize  $\phi^*$  to  $\phi^{n+\frac{1}{2}}$ 
9:   end if
10: end for
11: for all interfaces do
12:   Synchronize  $\phi^{n+\frac{1}{2}}$  from master to slaves
13: end for

```

Some remarks on this algorithm can be made: 1) The communication between master and slaves is reduced by the local marker definition. If the master and slave coincide, no communication is needed at all. For an interface, there is only communication between the master and its slaves, so all other PEs are not receiving/sending any messages. 2) By statically assigning interfaces to master PEs, an optimal load balancing is achieved. The interfaces are always evenly distributed over all available PEs. Therefore, when one PE physically holds all interfaces, this does not increase computational time much.

2.3.2 Deflation method

The deflation method was introduced independently by Nicolaides [103] and Dostál [37] and is a two-level preconditioned Conjugate Gradient (2L-PCG) method whose two-level preconditioner is based on a deflation technique. Tang [150] applied the method to bubbly flows, where the density ratio is relatively large. It was shown that 2L-PCG methods are less sensitive to the density ratio, the number of bubbles

and the grid resolution compared to standard PCG methods. The deflation method reduces the number of iterations of the CG solver, and is therefore an effective method for multiphase flow simulations.

Deflated PCG method

Since M^{-1} often only treats the largest eigenvalues of A effectively, there are still unfavorable small (nearly zero) eigenvalues left in the spectrum of $M^{-1}A$, $\sigma(M^{-1}A)$. The deflation method effectively treats these small eigenvalues by projecting them to zero. Since the convergence does not depend on zero eigenvalues [67], the effective condition number κ is reduced, and the convergence is accelerated. The deflated version of Eq. (2.26) is given by

$$M^{-1}PAx = M^{-1}Pb, \quad (2.29)$$

where P is the deflation matrix which is defined by

$$P := I - AQ, \quad Q := ZE^{-1}Z^T, \quad E := Z^T AZ. \quad (2.30)$$

Here Z is the so-called deflation-subspace matrix, which columns in the ideal case consists of eigenvectors associated with the most unfavorable eigenvalues. The Galerkin matrix, E , is also known as the coarse matrix. The 2L-PCG method is implemented as shown in Algorithm 4.

Algorithm 4 Deflated PCG algorithm

- 1: Select x_0 . Compute $r_0 = b - Ax_0$ and $\hat{r}_0 = Pr_0$,
 - 2: solve $My_0 = \hat{r}_0$ and set $p_0 := y_0$.
 - 3: **for** $j := 0, \dots$, until convergence **do**
 - 4: $\hat{w}_j := PAp_j$
 - 5: $\alpha_j := \frac{(\hat{r}_j, y_j)}{(p_j, \hat{w}_j)}$
 - 6: $\hat{x}_{j+1} := \hat{x}_j + \alpha_j p_j$
 - 7: $\hat{r}_{j+1} := \hat{r}_j - \alpha_j \hat{w}_j$
 - 8: Solve $My_{j+1} = \hat{r}_{j+1}$
 - 9: $\beta_j := \frac{(\hat{r}_{j+1}, y_{j+1})}{(\hat{r}_j, y_j)}$
 - 10: $p_{j+1} := y_{j+1} + \beta_j p_j$
 - 11: **end for**
 - 12: $x_{it} := Qb + P^T x_{j+1}$
-

Selection of deflation vectors

The most effective deflation vectors are the eigenvectors corresponding to the smallest nonzero eigenvalues of A . Unfortunately, these eigenvectors are usually expensive to compute and often dense, making the deflation method inefficient in use. For practical use, deflation vectors should be easy to construct, sparse and a good

approximation of the unfavorable eigenspace. Additionally, the deflation method should be well parallelizable, which is discussed in more detail in Section 2.3.2.

Tang [150] compared different known approaches for the choice of deflation vectors, like approximated eigenvector, recycling, subdomain and multigrid deflation vectors. He concluded that only subdomain deflation vectors satisfy all required criteria. Therefore, in this work only subdomain deflation is used. In subdomain deflation the computational domain is divided into several subdomains, where each subdomain corresponds to one deflation vectors. The deflation vector itself consists of ones in the interior of the subdomain and zeros elsewhere. This results in piecewise-constant, disjoint and orthogonal deflation vectors.

Each droplet corresponds to an unfavorable small eigenvalue [160], which is related to the density ratio of the dispersed and continuous phase. An increase of the number of droplets lead to more of these small eigenvalues, which slows down the convergence of the iterative solve. Therefore, the number of subdomains should at least be larger than the number of droplets. The optimal number of deflation vectors k can be approximated by

$$\psi = \left(\frac{n}{k}\right)^{1/3}, \quad (2.31)$$

where n is the total number of cells, and ψ should be a value between 5 and 10 [150]. In general, more deflation vectors will lead to a faster convergence of the iterative process. However, the costs per iteration will also be higher, due to the larger Galerkin system.

Efficiency and implementation

It is essential to have a good implementation of the deflation method in order to obtain an efficient method. As described in Section 2.3.2, a larger number of deflation vectors usually leads to faster convergence, but increases the costs per iteration. This increase in computational costs is a direct result of a larger Galerkin matrix, E , that should be solved during step 3 of Algorithm 4. Algorithm 5 [150] shows the procedure for computing $P\mathbf{y}$, where P is the deflation matrix and \mathbf{y} an arbitrary vector.

Algorithm 5 Computation of $P\mathbf{y}$

- 1: Compute $\mathbf{y}_1 := Z^T \mathbf{y}$
 - 2: Solve $E\mathbf{y}_2 := \mathbf{y}_1$
 - 3: Compute $\mathbf{y}_3 := (AZ)\mathbf{y}_2$
 - 4: Compute $P\mathbf{y} := \mathbf{y} - \mathbf{y}_3$
-

The matrices AZ and E can be constructed in a sparse way, which reduces the computational time significantly. Tang [150] defined two DICCG methods, which varied on the way to solve the Galerkin system: DICCG₁ (direct solve) and DICCG₂ (iterative solve). It was shown that DICCG₁ is efficient for small Galerkin systems ($k < 10^3$), and DICCG₂ for large Galerkin systems ($k > 10^3$). Since the difference

for $k < 10^3$ values was almost negligible, here only DICCG2 is used. If DICCG2 is used, an inner-outer iterative process is obtained. Here DICCG is used in the outer-iteration process and ICCG in the inner-iteration process. This leads to two different termination criteria, which can be written as

$$\delta_{\text{inner}} = \omega \delta_{\text{outer}}, \quad \omega > 0. \quad (2.32)$$

Following Tang [150],

$$\omega = 10^{-2}, \quad (2.33)$$

is used, which assures an accurate solution of the Galerkin system and convergence of the outer iteration process.

2.3.3 Three-dimensional domain decompositioning

Three-dimensional domain decompositioning leads to smaller communication messages between processors, and therefore less communication time. This can be shown by assuming a N^3 non-periodic computational domain, which is distributed over $P = P_x \cdot P_y \cdot P_z$ processors. Here N is the grid size in one spatial direction, and P_x , P_y , and P_z are the number of processors in each spatial direction. If this domain is only parallelized in one spatial direction, the total message size to synchronize between processors scales with

$$M_{1D} \propto (P - 1) \cdot N^2. \quad (2.34)$$

However, if this domain is parallelized in all three spatial directions, the total message size to synchronize between processors scales with

$$M_{3D} \propto ((P_x - 1) + (P_y - 1) + (P_z - 1)) \cdot N^2, \quad (2.35)$$

where P_x , P_y , and P_z are all > 1 . For all three-dimensional parallelizations $M_{3D} \ll M_{1D}$ holds, which shows that communication times also decrease if a three-dimensional domain decompositioning is used.

2.4 NUMERICAL EXPERIMENTS

To quantify the results of the proposed modifications of the previous section, in this section a numerical experiment with the buoyant rise of a regular array of droplets in a periodic box is performed, see Figures 2.1 and 2.2. Both the density ratio ρ_1/ρ_0 and the viscosity ratio μ_1/μ_0 are fixed to 0.1, and the Weber number is set to 2.0. The size of the periodic box is chosen as $6D \times 8D \times 8D$, where D is the droplet diameter. Each droplet is placed in the center of a $2D \times 2D \times 2D$ box, making it a $3 \times 4 \times 4$ droplet distribution. If less than 48 droplets are simulated, some of these $2D \times 2D \times 2D$ boxes are left empty.

Coyajee [26] investigated the influence of the grid resolution on the results for three different D/h ratios, where h is the cell size. Therefore, here results are shown for the same three D/h ratios: 16, 24, and 32, which correspond to respectively

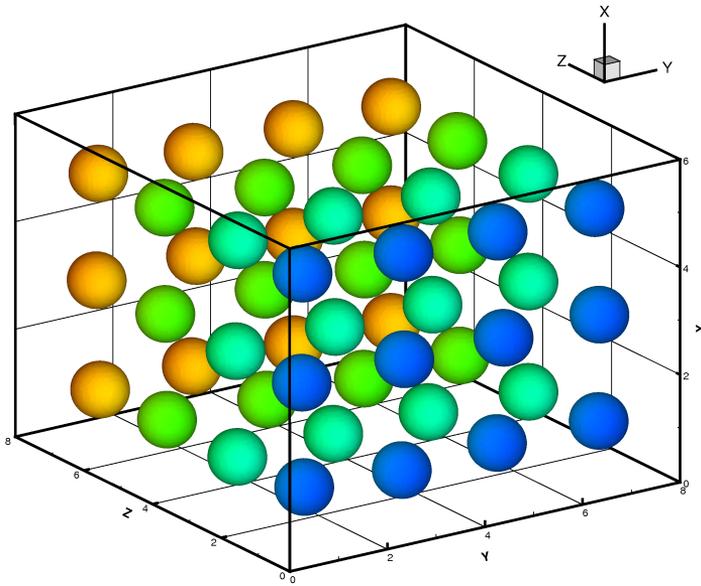


Figure 2.1: Initial droplet distribution for the buoyant rise of 48 droplets in a regular array.

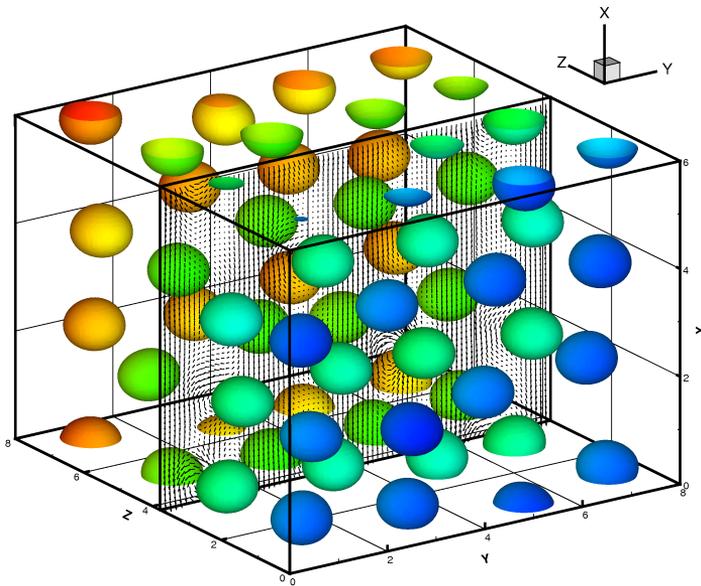


Figure 2.2: Droplet distribution and a velocity vector slice for the buoyant rise of 48 droplets in a regular array after 11000 timesteps.

the following grid sizes: $96 \times 128 \times 128$, $144 \times 192 \times 192$, and $192 \times 256 \times 256$. The time step limitation is given by Eq. (2.12), which leads to $\Delta t \approx 0.141 \cdot (h/D)^{3/2}$. All computations are performed on a 48-core AMD Opteron 6172 (2.1 GHz) with a total shared memory capacity of 64 GB. Moreover, the codes are compiled by Intel's compiler ifort (version 12.0.0) and parallelized by MPICH2 (version 1.3). To ensure that results are not accidental, averaging over the first 100 time steps is performed.

2.4.1 Effect of local multiple marker method

In the original, global, implementation of the multiple marker method of Coyajee and Boersma [27], the wall clock time to advect interfaces scales as

$$t_{\text{global}} \propto N_i \cdot \frac{M}{P_M}, \quad (2.36)$$

where N_i is the number of interfaces, M the number of grid cells in the narrow band for one interface, and P_M the number of processors which physically hold the marker. It is expected from Eq. (2.36) that the wall clock time to update the interfaces scales linearly with the number of interfaces N_i . This is also obtained for a test with an increasing number of interfaces on three different grid sizes, see Figure 2.3. It must be noted that not only M , but also P_M varies for the different grid sizes.

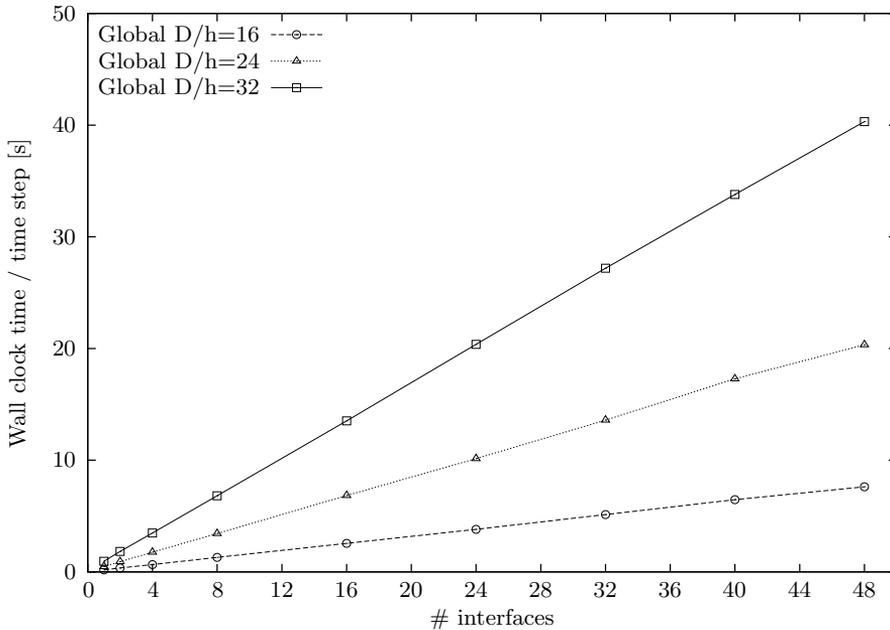


Figure 2.3: Wall clock time per time step to update interfaces with the global multiple marker method on a $48 \times 1 \times 1$ parallelization.

For the local method, the wall clock time to advect interfaces scales as

$$t_{\text{local}} \propto \left\lceil \frac{N_i}{P} \right\rceil \cdot M, \quad (2.37)$$

where P is the total number of processors used, and $\lceil \dots \rceil$ the ceiling operator. Since in general $P_M \ll P$, for the advection times it follows that $t_{\text{global}} \gg t_{\text{local}}$. It is expected from Eq. (2.37) that the wall clock time to update the interfaces for a fixed M only depends on the maximum number of interfaces, N_i , per processor. This is also obtained for a test with an increasing number of interfaces on three different grid sizes, see Figure 2.4. The weak slope of the lines is the result of an increase in communication between the master and its slaves. This effect can be reduced by choosing a more appropriate parallelization, see Section 2.4.3.

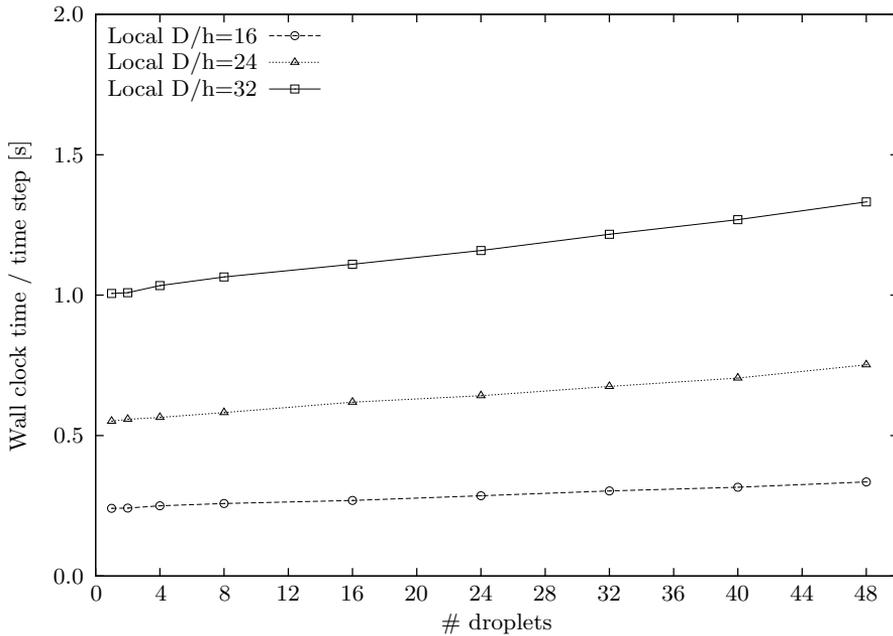


Figure 2.4: Wall clock time per time step to update interfaces with the local multiple marker method on a $48 \times 1 \times 1$ parallelization.

To see how the local multiple marker method scales with multiple interfaces per processor, a test is performed with 48 interfaces on a varying number of processors. Figure 2.5 shows, as expected from Eq. (2.37), it is equally expensive to update 48 interfaces on 24 as on 32 processors. Furthermore, from the spacing between the horizontal lines it becomes clear that the wall clock time per time step scales linearly with the maximum number of interfaces per processor.

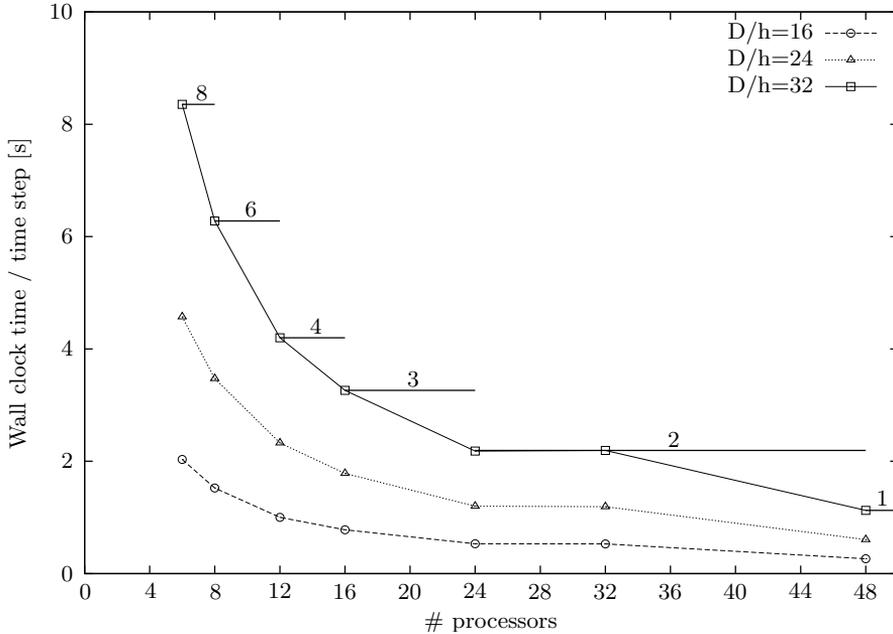


Figure 2.5: Wall clock time per time step to update 48 interfaces with the local multiple marker method. The horizontal lines indicate the maximum number of interfaces per processor.

2.4.2 Effect of deflation method

In theory, the deflation method should decrease both the number of iterations and wall clock time for the pressure solver. These effects can be seen in Figures 2.6 and 2.7 for an increasing number of interfaces on a $D/h = 32$ grid with a $3 \times 4 \times 4$ parallelization. For this parallelization, the effect of deflation can be summarized as follows: a reduction of the number of iterations with a factor of about 6, and a reduction of the wall clock time with a factor of about 4. From Figure 2.8 it is clear that the cost per iterations for the deflation method is higher than without the deflation method, which is a result of the additional systems to be solved. The large reduction in both the number of iterations and wall clock time for $N = 24$ and $N = 48$ are a direct result of the positioning of the interfaces in the domain. By favorable positioning of interfaces, the condition number, κ , of the system is kept low, which in turn makes the system easier to solve. However, such an advantageous distribution of interfaces is usually not maintained during simulations. Therefore, the deflation method is essential to remove unwanted fluctuations in the wall clock time and to speed up the pressure solver.

The deflation method has various adjustable parameters, where the most important ones are the shape of a deflation vector and the number of the deflation vectors. For $D/h = 32$ with a $3 \times 4 \times 4$ parallelization each processor holds $64 \times 64 \times 64$ cells. It is therefore expected that the optimal shape of a deflation vector is also a cube.

Figures 2.9 and 2.10 show respectively that both the number of iterations, and the wall clock time for cube shaped deflation vectors is the optimal choice. The lowest wall clock time is achieved by $4 \times 4 \times 4$ deflation, where the size of each deflation vector is $16 \times 16 \times 16$ cells. The total number of deflation vectors in that case is 3072, so DICCG2 is the best method to solve the Galerkin system. Tang [150] proposed by Eq. (2.31) that ψ should be a value between 5 and 10. Here $\psi = 16$ is obtained, which is considerably higher and indicates that the number of deflation vectors, k , is over-predicted by Eq. (2.31). A better approximation of the number of deflation vectors, k , is given by

$$k = \sqrt{n} \tag{2.38}$$

where n is the total number of cells. However, since this relation is empirically obtained, it still might be suboptimal.

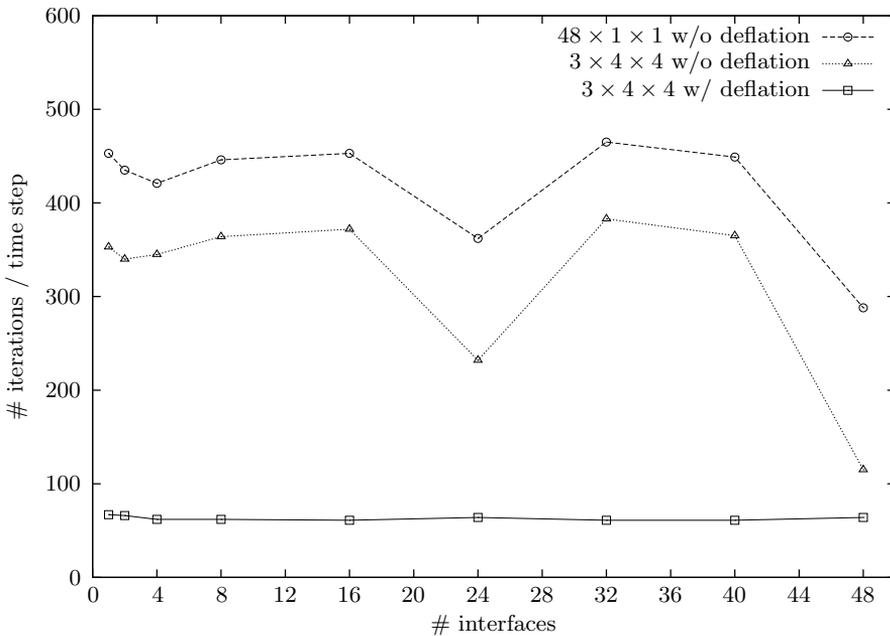


Figure 2.6: Number of iterations of the pressure solver per time step on a $D/h = 32$ grid for various deflation settings and parallelizations.

2.4.3 Effect of three-dimensional domain decomposition

By using a three-dimensional domain decomposition, the efficiency of both the local multiple marker method and the deflation method improve. Furthermore, also halo region synchronizations take less time, due to smaller message sizes. First, for the local multiple marker method the number of slaves is reduced, leading to smaller communication groups for the MPI process. This reduces possible wait time,

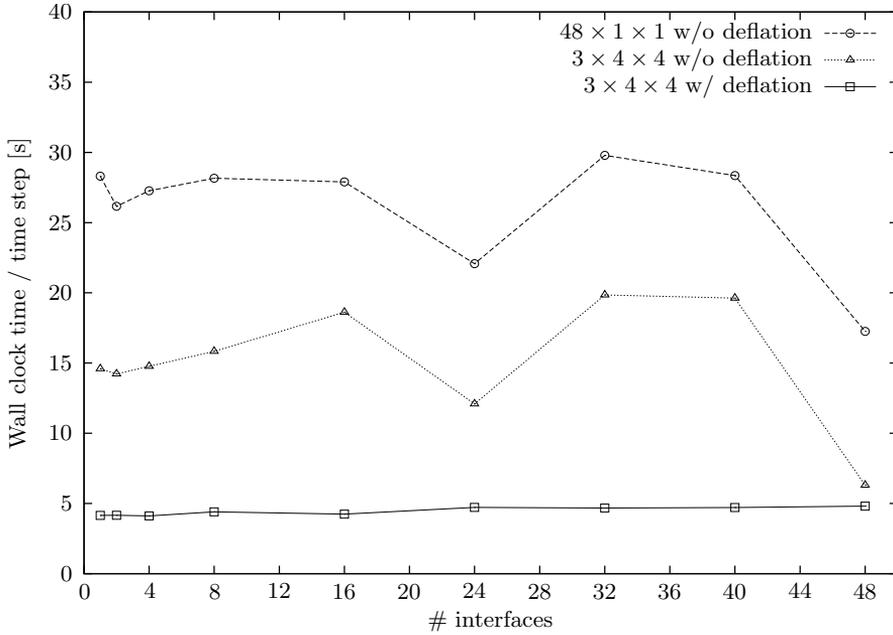


Figure 2.7: Wall clock time of the pressure solver per time step on a $D/h = 32$ grid for various deflation settings and parallelizations.

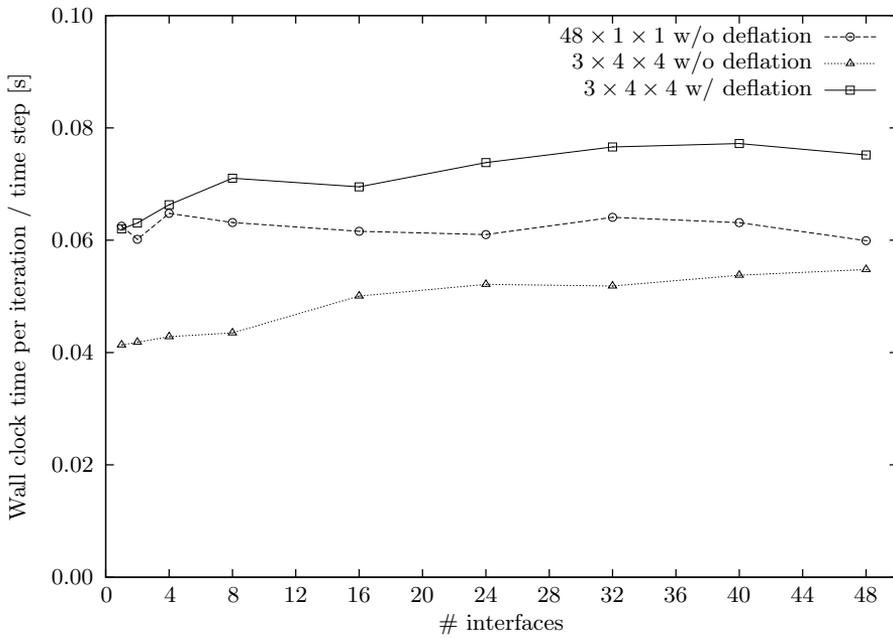


Figure 2.8: Wall clock time per iteration of the pressure solver per time step on a $D/h = 32$ grid for various deflation settings and parallelizations.

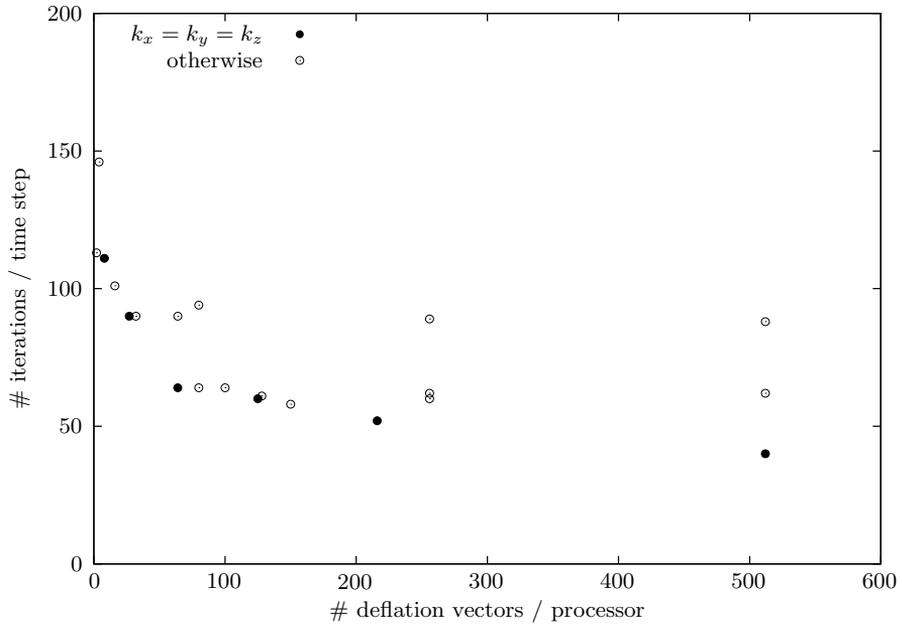


Figure 2.9: Number of iterations of the pressure solver per time step on a $D/h = 32$ grid for various shaped and sized deflation vectors.

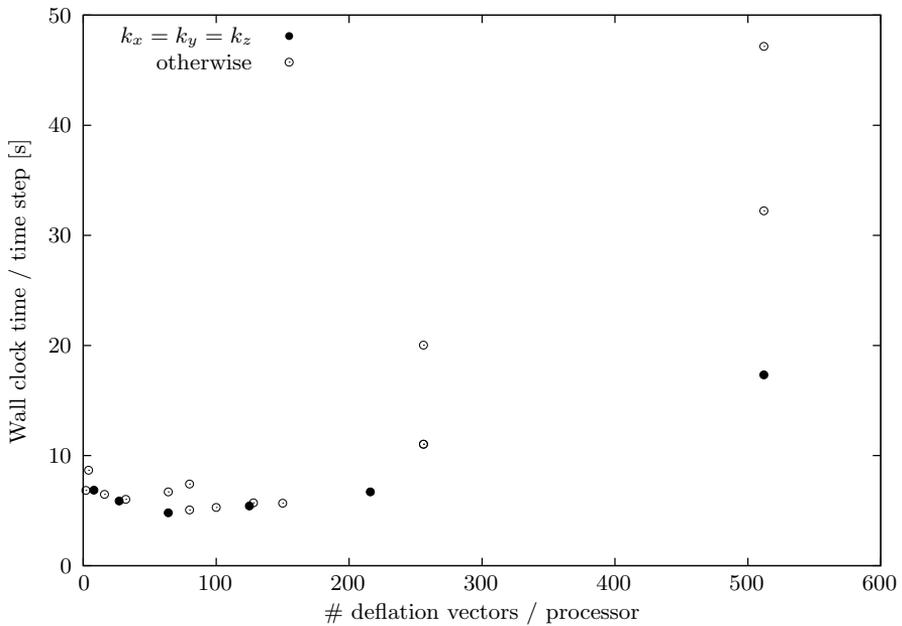


Figure 2.10: Wall clock time of the pressure solver per time step on a $D/h = 32$ grid for various shaped and sized deflation vectors.

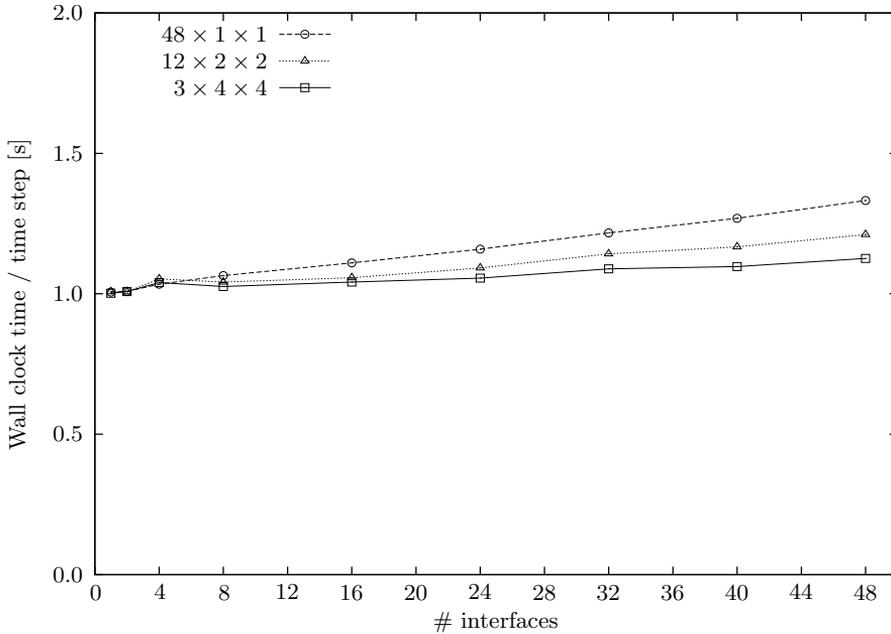


Figure 2.11: Wall clock time per time step to update interfaces with the local multiple marker method on a $D/h = 32$ grid for various parallelizations.

which can occur if one processor contributes to multiple masters. Figure 2.11 shows this effect for three different parallelizations. As expected, a $3 \times 4 \times 4$ parallelization is optimal, especially if the number of interfaces increases. Second, the default conditioning of the Poisson equation is improved, which leads to less iterations in the pressure solver without the use of deflation, see Figure 2.6. Figure 2.8 shows that also the amount of communication is reduced, since the wall clock time per iteration is reduced by using a three-dimensional parallelization. Again, a $3 \times 4 \times 4$ parallelization is optimal. Finally, the effect on other aspects of the code can be analyzed by subtracting the wall clock time spend in the pressure solver and for the interface update from the total wall clock time per time step, see Figure 2.12. By using the optimal $3 \times 4 \times 4$ parallelization, a speed up of about 1.5 can be achieved with respect to a $48 \times 1 \times 1$ parallelization.

2.4.4 Overall effect

Figure 2.13 shows the overall effect on the total wall clock time per time step for the original version with global multiple markers, no deflation, and a $48 \times 1 \times 1$ parallelization with respect to the improved version with local multiple marker, deflation, and a $3 \times 4 \times 4$ parallelization for three different grid sizes. Of course the speed up depends on the number of interfaces, but for 48 interfaces with $D/h = 32$ a speed up factor of more than 8 is already achieved. It must be noted, that the pressure solver

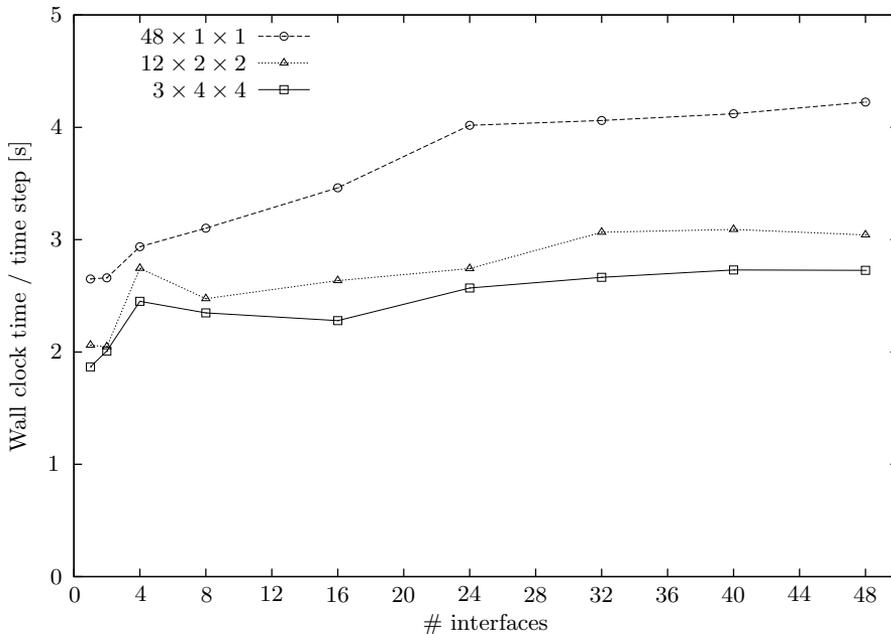


Figure 2.12: Wall clock time per time step without the pressure solver and interface update wall clock times on a $D/h = 32$ grid for various parallelizations.

of the original version for 48 droplets benefited from the advantageous distribution of the droplets. If this is not the case, an even higher speed up is expected.

2.5 CONCLUSIONS

To be able to perform DNS of a large number of droplets, a very accurate and efficient method is necessary. Accuracy was already achieved by using the multiple marker CLSVOF method [26, 27], but this implementation still lacked efficiency (especially if the number of droplets was increased). By combining the local multiple marker method with master/slave technique with the deflated PCG method and a three-dimensional domain decomposition this bottleneck is removed.

The local multiple marker method makes the interface update process less dependent on the number of droplets. Where the global multiple marker method scales linearly with the amount of interfaces, the scaling for the local multiple marker method only depends on the maximum number of interfaces per processor. The deflation method effectively removes unfavorable eigenvalues from the system to be solved. Therefore, the number of iterations can be reduced by a factor of about 6, and the wall clock time spend in the pressure solver can be reduced by a factor of about 4. These results scale very well with increasing grid sizes, increasing number of interfaces, and a larger density ratio. An alternative to the deflation method is the multigrid preconditioned conjugate gradient (MGCG) method [73, 152], which

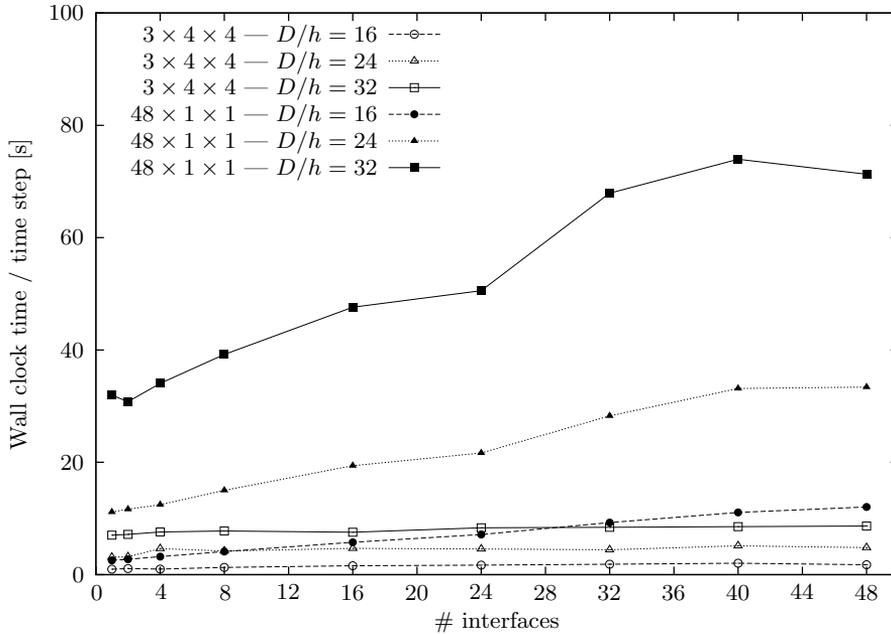


Figure 2.13: Wall clock time per time step on three grid sizes to indicate overall effects of the improved version ($3 \times 4 \times 4$ parallelization) with respect to the original version ($48 \times 1 \times 1$ parallelization).

is algebraically strongly related to deflated PCG [97]. The three-dimensional domain decomposition leads in general to smaller messages to be send, thereby reducing the wall clock time even further.

Overall, a speed up of about 8 is achieved for the buoyant rise of 48 droplets in a periodic box. Therefore, the improvements enable the accurate investigation of systems with a large number of droplets.



Every drop in the ocean counts

— Yoko Ono

Abstract *A multiple marker Coupled Level-Set/Volume-of-Fluid (CLSVOF) method for droplet-laden flows has been extended with a physical model to accommodate coalescence and breakup of the droplets. In the CLSVOF method each droplet has its own locally defined marker functions. This prevents the problem of numerical coalescence in conventional Level-Set and Volume-of-Fluid methods when two droplet interfaces are less than one grid cell apart from each other. In the present model coalescence is based on a computationally efficient film drainage model, which predicts if and when two colliding droplets will coalesce. If the contact time between two colliding droplets exceeds the predicted film drainage time, coalescence is numerically accomplished by merging the marker functions of the two separate droplets. Similarly, breakup of a droplet is accommodated by splitting the marker functions of the droplet. Simulations are shown for (near) head-on collision of two equal droplets at Weber numbers of 2.3, 9.33, 13.63, and 61.4, corresponding to four different collision regimes. The good agreement with the experimental data for each Weber number demonstrates the capability of the CLSVOF method to simulate coalescence and breakup with physical realism. A sensitivity study has shown a strong sensitivity of the collision dynamics on the prescribed film drainage time, while the influence of the spatial grid resolution appears to be small. This underlines the importance of accurate models for the film drainage time.*

3.1 INTRODUCTION

In many droplet-laden flows coalescence and breakup of droplets play an important role. Examples are the growth of rain droplets in clouds [2, 11, 50] and the breakup of droplets in spray combustion systems [4, 12, 65]. Such flows are typically turbulent. The interaction between the droplets and the surrounding turbulent carrier flow is complex and still not well understood, especially when the droplet concentration is not dilute [6, 155]. Visibility constraints hamper detailed measurements of the velocity field within dense droplet-laden flows. Direct Numerical Simulation (DNS) provides an alternative to obtain detailed insight in the structure and dynamics of such flows, although computationally demanding.

‡ This chapter has been published in the Journal of Computational Physics [76]

Popular methods for DNS of droplet-laden flows are front-tracking and front-capturing methods. In front-tracking methods marker particles are used for an explicit discrete approximation of the interface [153]. In front-capturing methods the interface is implicitly represented by either an isosurface of a smooth phase indicator (Level-Set) function [34, 56, 151] or by the local volume fraction of the dispersed phase (Volume-of-Fluid function) [112, 127]. Changes in the interface topology are automatically dealt within front-capturing methods, while the front-tracking methods require regular re-meshing of the marker particles on the interface. Front-capturing methods seem therefore to be better suited for studying coalescence/breakup problems [154]. In this work we make use of the multiple marker Coupled Level-Set/Volume-of-Fluid (CLSVOF) method of Coyajee and Bowersma [27] and modified by Kwakkel et al. [75], based on the original CLSVOF method introduced by Sussman and Puckett [143]. The CLSVOF method combines the sharp interface representation of the Level-Set method with the mass-conserving interface representation of the Volume-of-Fluid method.

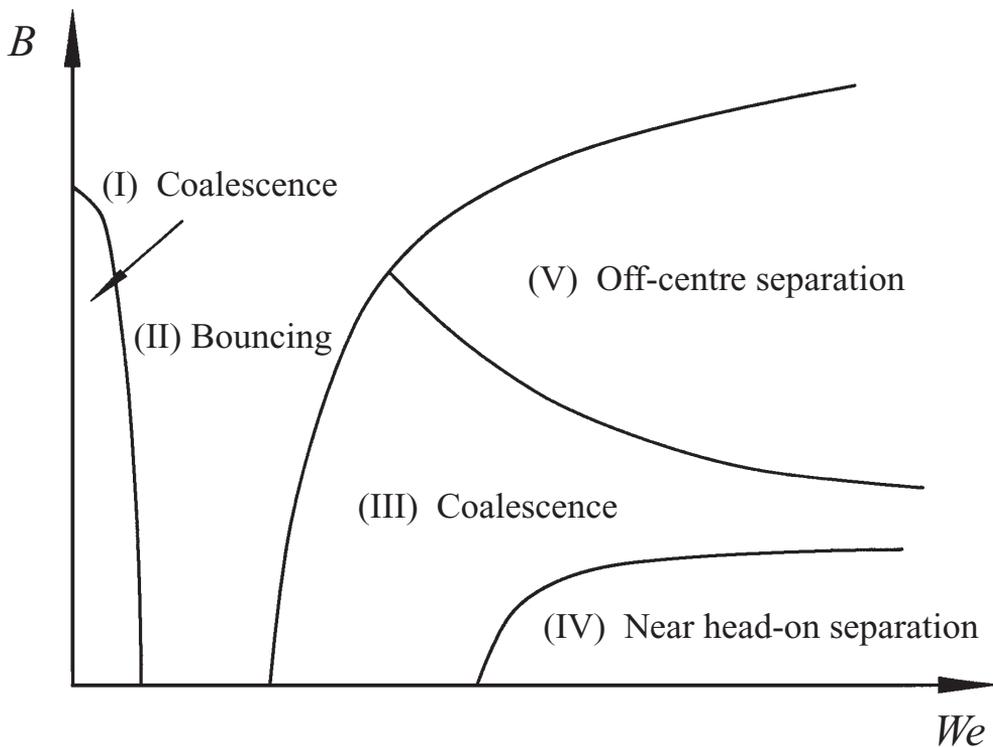


Figure 3.1: Schematic of various collision regimes of two equal sized hydrocarbon droplets in 1 atm. air. The parameter B represents the obliqueness of the collision (see Fig. 3.2), We denotes the Weber number. This figure is reproduced from Qian and Law [121].

A wide range of length scales is involved in the film drainage process, varying from length scales of the external flow that exceed the droplet diameter down to the nanometer scale at which the van der Waals force becomes active. It is computationally infeasible to capture this wide range of scales with a fixed-grid method, not even when excessive grid refinement would be applied. This means that in a fixed-grid method the final stage of film drainage will eventually always take place at the subgrid scale.

In conventional Level-Set (LS) and Volume-of-Fluid (VOF) methods two colliding droplets will automatically coalesce when their interfaces are sufficiently close to each other, typically at a distance on the order of the dimension of a grid cell [106, 116]. This is an artifact of these methods, commonly referred to as numerical coalescence, and originates from the numerical representation of separate droplets by a single LS and/or VOF function. In some cases, such as the head-on collision between two equal sized droplets, coalescence can be controlled by manipulating the boundary conditions at the symmetry plane in between the two droplets [64, 79]. However, such an approach is not applicable to droplet-laden flows in general.

To avoid numerical coalescence, Coyajee and Boersma [27] extended the CLSVOF method of Sussman and Puckett [143] with a multiple marker approach: every droplet is represented by its own LS and VOF function. The disadvantage of this method is, however, that it prohibits coalescence of droplets at all. In this paper we present an extension of this method in which the possibility of coalescence is reintroduced by combining it with a detailed film drainage model for the thin film in between two colliding droplets. From this film drainage model it can be determined if two colliding droplets will eventually coalesce or bounce. When coalescence is predicted, the separate marker (LS/VOF) functions of the two droplets are merged. Such an approach was already suggested by the group of Tryggvason [107, 155].

Several coalescence models have been developed during the past decades [80]: critical velocity models [78], energy models [62, 136], and film drainage models [22, 30]. In principle all these models could be used to predict/describe coalescence, but here the film drainage model is used. The advantage of this class of models is that it provides also an estimate of the total film drainage time prior to coalescence. If the required film drainage time is on the order or larger than the time scales of the external flow (e.g., the time scales of turbulence and shear), then the droplets may have been separated from each other before coalescence can actually become active. In the present study we make use of the film drainage model of Zhang and Law [166]. This model is able to predict the coalescence-bouncing-coalescence transition for increasing Weber number as illustrated in Figs. 3.1 and 3.2. Furthermore, it also provides an estimate of the required film drainage time prior to coalescence. Finally, this model is computationally efficient, which makes the implementation within the multiple marker CLSVOF method computationally feasible.

The goal of this research is the extension of our multiple marker CLSVOF method [27, 75] for droplet-laden flows with models to accommodate coalescence and breakup of droplets without significant loss of computational efficiency. The focus will be on coalescence, for which we make use of a film drainage model. Breakup is considered as well, since in the framework of the multiple marker approach breakup

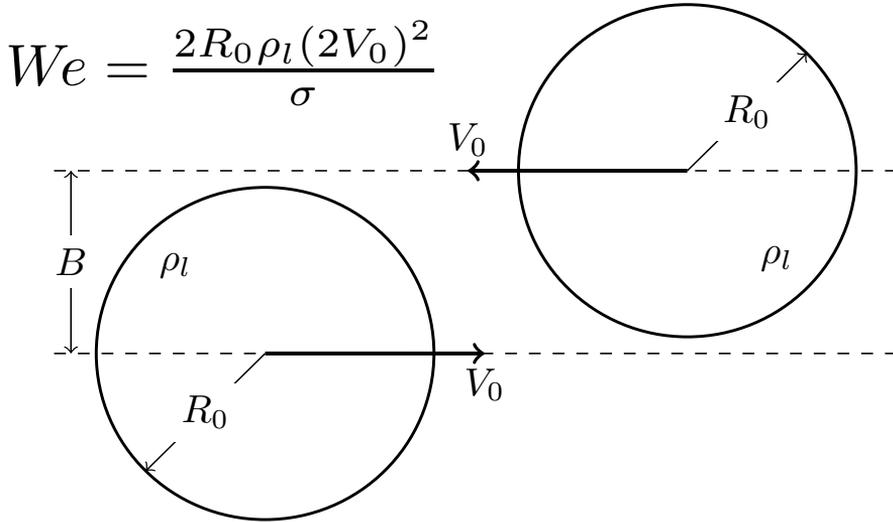


Figure 3.2: Nomenclature of a general binary droplet collision, where B is the impact parameter, V_0 the droplet velocity, R_0 the droplet radius, ρ_l the dispersed phase density and σ the surface tension coefficient. The Weber number We , which is a measure of inertial forces over surface tension forces, strongly influences the collision outcome, see Fig. 3.1. An impact parameter $B = 0$ corresponds to a pure head-on collision.

requires to split the marker of a parent droplet into separate markers for the child droplets. The structure of this paper is as follows. The details of the numerical method are described in Section 3.2. The extension of this method with a coalescence and breakup model is detailed in Section 3.3. In Section 3.4 simulation results are shown and compared with experimental data for the (near) head-on collision between two equal sized droplets at various Weber numbers corresponding to different collisions regimes. Furthermore, the influence of the prescribed film drainage time and the spatial grid resolution on the collision dynamics is investigated. This is followed by a summary of the main conclusions and a discussion in Section 3.5.

3.2 ORIGINAL NUMERICAL METHOD

3.2.1 Governing equations

The governing equations for two-phase flows of two incompressible, immiscible and Newtonian fluids are given by the Navier-Stokes equations for each phase:

$$\nabla \cdot \mathbf{u} = 0, \quad (3.1a)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{u}\mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \left(\mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right) \right) + \mathbf{g}, \quad (3.1b)$$

where \mathbf{u} is the velocity vector, p is the pressure, \mathbf{g} is the gravitational acceleration, and ρ and μ are the local mass density and the dynamic viscosity of the fluid, respectively. The mass density and viscosity are assumed constant within each phase and denoted by ρ_0, μ_0 in the continuous phase (phase '0') and ρ_1, μ_1 in the dispersed phase (phase '1').

Assuming that the two fluids are immiscible, the governing equations for each fluid are coupled with each other through the following conditions at the interface Γ between the two fluids:

$$[\mathbf{u}]_{\Gamma} = \mathbf{0} \quad (3.2a)$$

$$\left[-p\mathbf{n} + \mu \left(\nabla\mathbf{u} + \nabla\mathbf{u}^T \right) \cdot \mathbf{n} \right]_{\Gamma} = -\sigma\kappa_{\Gamma}\mathbf{n} \quad (3.2b)$$

where \mathbf{n} is the unit normal at Γ pointing from phase 0 to phase 1, $[\cdot]_{\Gamma}$ represents the jump in a quantity across the interface from phase 0 to phase 1 (i.e., the value of a quantity in phase 1 minus its value in phase 0), $\kappa_{\Gamma} \equiv -\nabla \cdot \mathbf{n}$ is the interface curvature and σ is the surface tension coefficient.

The interface between the two phases can be described with help of the Level-Set function ϕ , which we define as the signed closest distance towards the interface and which is negative in phase 0 and positive in phase 1. The position of the interface, Γ , can then be defined as the zero level of ϕ : $\Gamma(t) \equiv \{\mathbf{x} | \phi(\mathbf{x}, t) = 0\}$. Furthermore, the fluid mass density and the viscosity in Eq. (3.1b) can be written in the following forms:

$$\rho = \rho_0 (1 - H(\phi)) + \rho_1 H(\phi), \quad (3.3a)$$

$$\mu = \mu_0 (1 - H(\phi)) + \mu_1 H(\phi), \quad (3.3b)$$

where $H(\phi)$ is the Heaviside function equal to 0 for $\phi \leq 0$ and 1 for $\phi > 0$.

Because we assume that the fluids are immiscible, the interface Γ is a material property of the flow and can be tracked from solving the following equation for ϕ :

$$\frac{\partial\phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0, \quad (3.4)$$

where we used that for incompressible flows the velocity is divergence free. Note that Eq. (3.4) holds strictly speaking only for the position of the interface, that is $\phi = 0$.

3.2.2 Time integration and spatial discretization

The governing equations, Eqs. (3.1a) and (3.1b), are integrated in time with a standard pressure correction method. In symbolic notation this scheme reads as follows [27]:

$$\begin{aligned} \mathbf{u}^* = & \mathbf{u}^n + \Delta t \left(-\frac{3}{2}\mathcal{A}(\mathbf{u}^n) + \frac{1}{2}\mathcal{A}(\mathbf{u}^{n-1}) \right) \\ & + \frac{\Delta t}{\rho^{n+\frac{1}{2}}} \left(-\mathcal{G}p^{n-\frac{1}{2}} + \mathcal{D}_{\text{imp}}(\mathbf{u}^*) + \mathcal{D}_{\text{exp}}(\mathbf{u}^n) \right) \\ & + \Delta t \mathbf{g}, \end{aligned} \quad (3.5a)$$

$$\text{DIV} \left(\frac{\mathcal{G}p^*}{\rho^{n+\frac{1}{2}}} \right) = \frac{1}{\Delta t} \text{DIV}(\mathbf{u}^*), \quad (3.5b)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho^{n+\frac{1}{2}}} \mathcal{G}p^*, \quad (3.5c)$$

$$p^{n+\frac{1}{2}} = p^{n-\frac{1}{2}} + p^*, \quad (3.5d)$$

where \mathcal{A} is the discrete form of $\nabla \cdot \mathbf{u}\mathbf{u}$, \mathcal{G} is the discrete gradient operator, \mathcal{D}_{imp} is the discrete form of $\nabla \cdot (\mu \nabla \mathbf{u})$, \mathcal{D}_{exp} is the discrete form of $\nabla \cdot (\mu \nabla \mathbf{u}^T)$, DIV is the discrete divergence operator, \mathbf{u}^* is the prediction velocity, p^* is the correction pressure, $\rho^{n+\frac{1}{2}} = \rho(\phi^{n+\frac{1}{2}})$ is the mass density computed from Eq. (3.3a), Δt is the computational time step and the superscript n denotes time level n . The advection and diffusion term are integrated in time with the second-order Adams-Bashforth and the Crank-Nicolson scheme [164], respectively. In both the explicit and the implicit diffusion term the viscosity is evaluated at $n + \frac{1}{2}$.

Eq. (3.5a) for \mathbf{u}^* can be rewritten in matrix form. The resulting equation is solved with an Incomplete Cholesky preconditioned Conjugate Gradient (ICCG) method [51]. Similarly, Eq. (3.5b) for p^* is solved with a deflated ICCG method where the preconditioning is extended with a deflation method for reason of computational efficiency [75, 150]. The computational time step Δt is dynamically adjusted in time and computed from restrictions for numerical stability based on the effects of advection, surface tension and gravity [9, 26].

Eqs. (3.5a) to (3.5d) are discretized in a three-dimensional rectangular domain on a fixed, uniform and Cartesian grid with the finite-volume method. The computational grid is fully staggered in space [54] with the velocity components defined on the cell faces and the pressure, density, viscosity and marker functions defined in the cell centers. Spatial derivatives are estimated from the second-order central-differencing scheme.

The viscosity is regularized by replacing the Heaviside function $H(\phi)$ in Eq. (3.3b) by the following continuous function $H_\alpha(\phi)$ [21]:

$$H_\alpha(\phi) = \begin{cases} 0 & \text{if } \phi < -\alpha \\ \frac{1}{2} \left(1 + \frac{\phi}{\alpha} + \frac{1}{\pi} \sin \left(\frac{\pi \phi}{\alpha} \right) \right) & \text{if } |\phi| \leq \alpha \\ 1 & \text{if } \phi > \alpha \end{cases} \quad (3.6)$$

where $\alpha = 3h/2$ with h the uniform mesh width; the jump in the viscosity is thus smeared out over three mesh widths. Kang et al. [69] showed that when the viscosity is regularized, the velocity gradient becomes continuous across the interface between two phases. Consequently, the jump in the normal stress across the interface as given by Eq. (3.2b) reduces to an explicit jump in the pressure across the interface:

$$[p]_{\Gamma} = \sigma \kappa_{\Gamma}. \quad (3.7)$$

The above explicit jump condition for the pressure over the interface can be easily implemented in the discretization of the spatial pressure gradient with the Ghost-Fluid method [82]. Consider a pressure gradient in the streamwise direction x at position $i + \frac{1}{2}$ (in between the cell centers at i and $i + 1$) near an interface. This pressure gradient is discretized as follows:

$$\left. \frac{\partial p}{\partial x} \right|_{i+\frac{1}{2}} = \frac{p_{i+1} - p_i - \beta [p]_{\Gamma}}{h}, \quad (3.8)$$

where $\beta = +1$ if $\phi_i \leq 0$ and $\phi_{i+1} > 0$, $\beta = -1$ if $\phi_i > 0$ and $\phi_{i+1} \leq 0$, and $\beta = 0$ in all other cases.

Unlike the viscosity, the mass density is not regularized. The spatial discretization of Eq. (3.5b) requires the values of $1/\rho$ at the cell faces, while ρ is defined in the cell centers. In the vicinity of an interface these values are computed from the following weighted harmonic average [82]:

$$\frac{1}{\rho_{i+\frac{1}{2}}} = \frac{1}{\rho_i \theta + \rho_{i+1} (1 - \theta)}, \quad (3.9)$$

where $\theta = |\phi_i| / (|\phi_i| + |\phi_{i+1}|)$ is a function that varies between 0 and 1.

3.2.3 Multiple marker CLSVOF method

The advection of the Level-Set function given by Eq. (3.4) does not guarantee that the enclosed mass (or volume) is conserved. In the Coupled Level-Set/Volume-of-Fluid (CLSVOF) method [143] the conservation of mass is improved by combining the Level-Set (LS) method with the Volume-of-Fluid (VOF) method. In the VOF method the phases are represented by the Volume-of-Fluid function ψ , which is defined as the volume fraction of the dispersed phase (phase 1) in a computational grid cell, $\psi \equiv (1/h^3) \int_{h^3} H(\phi(\mathbf{x}, t)) dV$ with h^3 the volume of the grid cell. Conservation of mass requires that:

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\mathbf{u}\psi) = 0. \quad (3.10)$$

The LS and the VOF function are defined in the cell centers and at time half levels. The CLSVOF method consists of two steps [143]:

1. Advection of the LS and the VOF function. Given the velocity \mathbf{u}^{n+1} as obtained from Eq. (3.5c) and given $\psi^{n+\frac{1}{2}}$ and $\phi^{n+\frac{1}{2}}$ from the previous time

step, Eqs. (3.4) and (3.10) are integrated in time with a second-order accurate conservative operator split advection scheme [120, 143] to obtain $\psi^{n+\frac{3}{2}}$ and $\phi^{n+\frac{3}{2}}$. The discrete flux $\mathbf{u}\psi$ in Eq. (3.10) is computed from a piecewise linear reconstruction of the interface within a grid cell, where the co-advected LS function is used to obtain the normal vector to the reconstructed interface.

2. Correction of the advected LS function for conservation of mass and reinitialization of the corrected LS function. The correction is determined from a piecewise linear reconstruction of the interface in the cells cut by the interface, where the yet uncorrected LS function is used to obtain the normal vector to the reconstructed interface and the VOF function is used to determine the intercept of the linear reconstruction. Next, the corrected LS function is reinitialized over a width of 4 grid cells on either side of the interface by assigning it to the signed closest distance to the reconstructed interface. The signed closest distance is assigned by geometric reconstruction, where the VOF value and interface normal vector are used to reconstruct the interface. The signed closest distance is determined between each grid point and this reconstructed interface.

The interface curvature can be computed either from the LS or the VOF function. We compute it from the VOF function using the height-function method [26, 46, 55, 141]. Coyajee and Boersma [27] demonstrated that this method is second-order accurate.

The original CLSVOF method can not distinguish between different interfaces within the same grid cell. Therefore, when two droplets approach each other closely they will automatically coalesce, which is commonly referred to as numerical coalescence. In order to avoid this problem Coyajee and Boersma [27] extended the CLSVOF method with a multiple marker approach in which each interface has its own marker (LS/VOF) functions. In case of a two-phase flow with N dispersed droplets, the number of LS/VOF functions is then equal to N . Material properties are calculated from a common LS function ϕ^c to all interfaces as $\rho = \rho(\phi^c)$ and $\mu = \mu(\phi^c)$ with ϕ^c defined as:

$$\phi^c(\mathbf{x}, t) = \max(\phi^1(\mathbf{x}, t), \dots, \phi^N(\mathbf{x}, t)). \quad (3.11)$$

Furthermore, Eq. (3.8) for the discretization of the pressure gradient is generalized to account for the possible presence of multiple interfaces between grid nodes i and $i+1$:

$$\left. \frac{\partial p}{\partial x} \right|_{i+\frac{1}{2}} = \frac{p_{i+1} - p_i - \sum_{m=1}^N \beta^m [p]_{\Gamma}^m}{h}, \quad (3.12)$$

where the superscript m denotes the m -th droplet, $[p]_{\Gamma}^m = \sigma \kappa_{\Gamma}^m$, $\beta^m = +1$ if $\phi_i^m \leq 0$ and $\phi_{i+1}^m > 0$, $\beta^m = -1$ if $\phi_i^m > 0$ and $\phi_{i+1}^m \leq 0$, and $\beta^m = 0$ in all other cases.

Kwakkel et al. [75] improved the computational efficiency of the multiple marker CLSVOF method of Coyajee and Boersma [27] by using local marker functions that are defined only in a finite rectangular box that captures the full droplet. The

box moves along with the droplet and the dimensions of the box are dynamically adjusted in time, dependent on the deformation of the droplet. This improvement strongly reduces the memory requirements of the multiple marker approach when the number of droplets becomes very large.

3.2.4 *Parallelization*

The numerical algorithm detailed in the previous subsections is coded in Fortran with the Message Passing Interface (MPI) library extension for parallel execution on multi-processor machines with distributed memory [75]. For the parallelization of the Navier-Stokes equations a standard domain decomposition method in three dimensions is used. Each subdomain is assigned to a separate process of the parallelization and has the same number of grid cells to optimize load balancing. For the parallelization of the multiple marker CLSVOF method a master-slave method has been used. In this method the number of droplets (or interfaces in general) are as much as possible equally distributed over the parallel processes. The process to which a droplet is assigned at the start of simulation is referred to as the master of the droplet; this process remains the master of this droplet during the course of the whole simulation. The processes of the subdomains in which a droplet is physically located are referred to as the slaves of the droplet; at least one slave is assigned to a droplet, but more is possible. The LS and VOF functions of a droplet are advected, corrected, and reinitialized by the master of the droplet, which receives the required velocity field from the slaves of the droplet. Furthermore, the master computes the surface tension (interface curvature) from the advected VOF function with help of the height-function method. The advected LS function and the surface tension are then sent back to the slaves. The received LS function is used by the slaves to compute the local viscosity and density.

3.3 EXTENSION WITH COALESCENCE/BREAKUP MODEL

3.3.1 *Coalescence model*

Before two colliding droplets coalesce, first the film between the two droplets has to be drained. The film drainage speed is initially much higher than later on in the process of film drainage. This implies that the precise starting time of a collision, which definition is somewhat arbitrary, is unimportant for predicting the time at which coalescence takes place. Coalescence typically takes place if the film thickness h is $\mathcal{O}(100)$ nm. At this distance the van der Waals force comes into play, resulting in a quick rupture of the film. However, before this critical film thickness is reached, droplets could have lost contact due to conversion of surface energy into kinetic energy (bouncing) or external flow conditions (e.g., turbulence).

For the coupling of the multiple marker CLSVOF method with a coalescence model, we follow the conceptual framework of Chesters [22]. In this framework the flow is split into an internal and an external part, corresponding to, respectively,

the flow inside the drainage film and outside the film / around the droplets. The internal flow dynamics is modeled by the coalescence model with the external flow conditions as input. From the coalescence model it is determined if and when the colliding droplets will coalesce. The external flow is resolved by the CLSVOF method, which determines the number of droplet collisions, the (initial) interaction force/velocity of colliding droplets and the contact time between colliding droplets. Although in the multiple marker CLSVOF method the film between two colliding droplets is unresolved, at least in the later stage of film drainage, the effect on the overall droplet dynamics is expected to be small [45]. When the simulated film becomes thin compared to the dimensions of a grid cell, the pressure inside the unresolved film is represented by the nearest pressure nodes located within the droplets. This is in accordance with the physical expectation that for a nearly flat film the pressure inside the film is of the same order of magnitude as the pressure inside the droplets close to the interface. Thus, the CLSVOF method accurately captures the overall droplet dynamics, while the multiple marker approach prevents immediate coalescence of the droplets at close approach.

Film drainage model

In this work the recently introduced film drainage model of Zhang and Law [166] is used, which determines the outcome of a head-on collision of two identical droplets in a gaseous environment. Their model incorporates the most essential physics of a binary droplet collision: the deformation of the droplets, viscous dissipation of kinetic energy through internal droplet motion, the dynamics and rarefied nature of the gaseous film in between the droplets, and the van der Waals force during the last stage of drainage. From these principles, the model is able to predict the non-monotonic coalescence-bouncing-coalescence transition for increasing We , see Fig. 3.1. Besides the outcome of a collision, the model also provides an estimate of the film drainage time, t_d , which can be compared with the contact time, t_c , of the droplets (determined by the external flow). Moreover, the model can be solved efficiently, which makes the implementation of this model in the multiple marker CLSVOF method computationally feasible.

The deformed shape of a colliding droplet is approximated by a truncated sphere with radius $R(t)$ and a flattened disc-shaped interface of radius $a(t)$, see Fig. 3.3. The film thickness between the droplets is denoted by $h(t)$, and the distance between the center, O , of the spherical cap and the flat interface is denoted by $b(t)$. The distance between the center of mass, C_m , and the flat interface is denoted by $G(t)$. Concerning the truncated-sphere approximation of the droplets during the collision, we would like to make a few remarks. First, for collisions at high We , the rear side of the droplet does not remain spherical [65]. Second, the sharp transition between the spherical cap and the flat interface is in reality smoothed by surface tension [19]. Third, in reality the film in between the droplets does not remain flat, but it evolves into a dimpled shape [19]. However, it is expected that all these assumptions do not significantly influence the results [166].

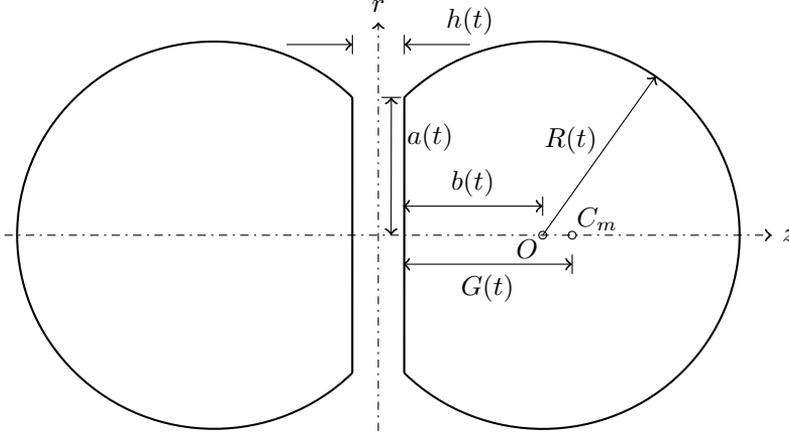


Figure 3.3: Configuration and parameters of the droplet-collision model of Zhang and Law [166] for head-on collision of two equal sized droplets.

To solve the evolution of the film thickness h in time a system of four ordinary differential equations (ODE) has to be solved. These equations originate from geometrical relations and the equations for conservation of mass, momentum and energy. The equations are made dimensionless with help of the impact velocity V_0 (defined as half the absolute value of the relative velocity between the two droplets) and the droplet radius R_0 prior to collision. The non-dimensional variables are denoted by a tilde. Differentiation in time is denoted by a prime. In non-dimensional form the equations read as follows [166]:

$$\tilde{a}' = - \left(\frac{\tilde{R} + \tilde{b}}{2\tilde{a}} \right) \tilde{b}', \quad (3.13a)$$

$$\tilde{R}' = - \left(\frac{\tilde{R} - \tilde{b}}{2\tilde{R}} \right) \tilde{b}', \quad (3.13b)$$

$$\tilde{b}'' = \frac{2(2\tilde{R} - \tilde{b})}{(\tilde{R} + \tilde{b})} \tilde{G}'' - \frac{3}{2\tilde{R}} (\tilde{b}')^2, \quad (3.13c)$$

$$\tilde{h}' = - \frac{4Re_l \tilde{h}^3}{9\lambda \tilde{a}^4} \left(\tilde{G}'' + \tilde{A}_H \frac{\tilde{a}^2}{\tilde{h}^3} \right) \Delta(Kn) - 2\tilde{\kappa}\tilde{h}, \quad (3.13d)$$

where in the last equation $Re_l \equiv 2\rho_l R_0 V_0 / \mu_l$ is the droplet Reynolds number where the subscript l denotes the dispersed (liquid) phase, $\lambda \equiv \mu_g / \mu_l$ is the gas-to-liquid viscosity ratio, $\tilde{A}_H \equiv A_H / 8\pi\rho_l V_0^2 R_0^3$ is the non-dimensional Hamaker constant, $\Delta(Kn)$ is a non-dimensional function [166] of the Knudsen number $Kn = \lambda_g / h$ with λ_g the mean free path of gas molecules, and $\tilde{\kappa} \equiv \tilde{a}' / \tilde{a}$ is the radial strain rate of the disk-shaped interface. The expression for \tilde{G}'' in Eqs. (3.13c) and (3.13d) is given by the following equation:

$$\tilde{G}'' = - \frac{36}{Re_l^3} \frac{\tilde{\Phi}_l}{\tilde{b}'} \frac{(2\tilde{R} - \tilde{b})}{(\tilde{R} + \tilde{b})} - \frac{24}{We} \frac{\tilde{R}'}{\tilde{b}'} (2\tilde{R} - \tilde{b}), \quad (3.14)$$

where $We \equiv 8\rho_1 V_0^2 R_0 / \sigma$ is the Weber number and $\tilde{\Phi}_l$ is the viscous dissipation rate of internal motion inside each droplet. Note that we don't need to integrate Eq. (3.14) in time; it is solely needed to compute \tilde{G}'' in Eqs. (3.13c) and (3.13d). The viscous dissipation rate is given by the following expression:

$$\begin{aligned} \tilde{\Phi}_l = & \left(\frac{21}{48} \tilde{R} \tilde{\eta}^2 - \frac{1}{24} \tilde{R}^2 \tilde{\eta}^3 + \frac{21}{96} \tilde{\eta} \right) e^{-2\tilde{\eta}(\tilde{R} + \tilde{b})} \\ & + \frac{1}{48} \left(\tilde{R}^2 - \tilde{b}^2 \right)^2 \tilde{\eta}^5 + \frac{1}{24} \left(\tilde{R}^2 - \tilde{b}^2 \right) \tilde{b} \tilde{\eta}^4 \\ & + \frac{1}{48} \left(23\tilde{R}^2 - 21\tilde{b}^2 \right) \tilde{\eta}^3 + \frac{21}{48} \tilde{b} \tilde{\eta}^2 - \frac{21}{96} \tilde{\eta}, \end{aligned} \quad (3.15)$$

where $\tilde{\eta} \equiv 1.503 (Re_l |\tilde{\kappa}|/2)^{1/2}$.

The four differential equations (3.13a) to (3.13d) are solved in time with the solver LSODE [122] for \tilde{a} , \tilde{R} , \tilde{b}' and \tilde{h} , respectively. An additional equation is solved for the variable b , given by the following geometrical relation:

$$\tilde{b} = \sqrt{\tilde{R}^2 - \tilde{a}^2}. \quad (3.16)$$

Zhang and Law [166] formulated the following initial conditions* for the system of equations (3.13a) to (3.13d) and (3.16):

$$\tilde{h}(0) = \tilde{h}_d \exp\left(-\delta \tilde{h}_d^{3/2} / \lambda\right), \quad (3.17a)$$

$$\tilde{a}(0) = [\tilde{h}(0)]^{1/2}, \quad (3.17b)$$

$$\tilde{b}(0) = [1 - \tilde{a}^2(0)]^{1/2}, \quad (3.17c)$$

$$\tilde{b}'(0) = -1, \quad (3.17d)$$

$$\tilde{R}(0) = 1, \quad (3.17e)$$

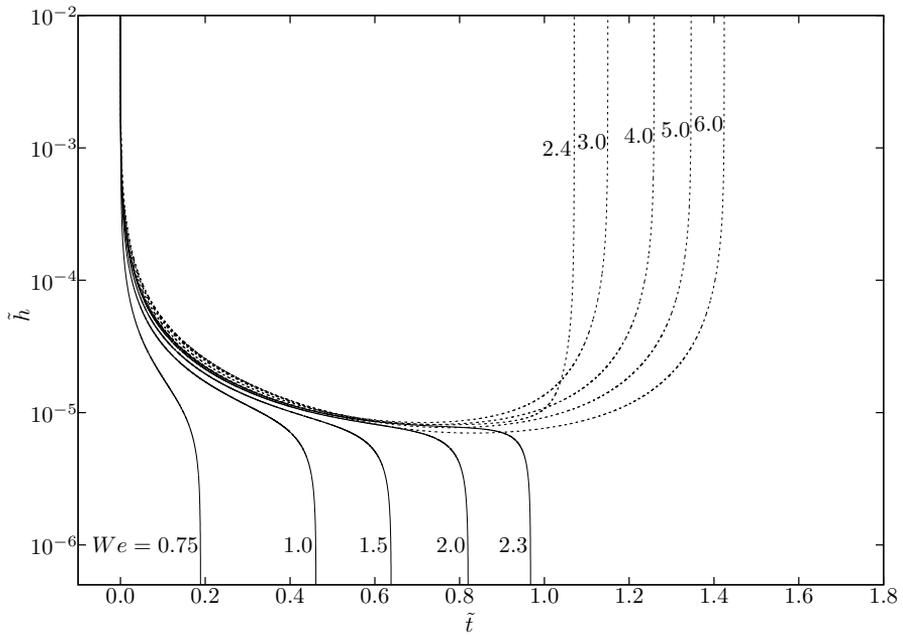
where $\tilde{h}_d = (3Ca/2)^{1/2}$ is the film thickness at the start of significant deformation of the droplets with $Ca = \mu_g V_0 / \sigma$ the Capillary number, and δ is an empirical coefficient.

Some deficiencies in the physics of this model should be noted. First, the assumption of a uniform film thickness results in a parabolic radial pressure distribution within the film. This does not match with existing more detailed film drainage models with a radially varying film thickness, which show a more or less constant radial pressure distribution within the film [71, 126]. Second, the empirical coefficient δ strongly influences the final outcome of the model. Here $\delta = 1.837$ is used for tetradecane droplets in air in order to get the best match with experiments over a range of We , while $\delta = 1.3$ was used by Zhang and Law [166][†].

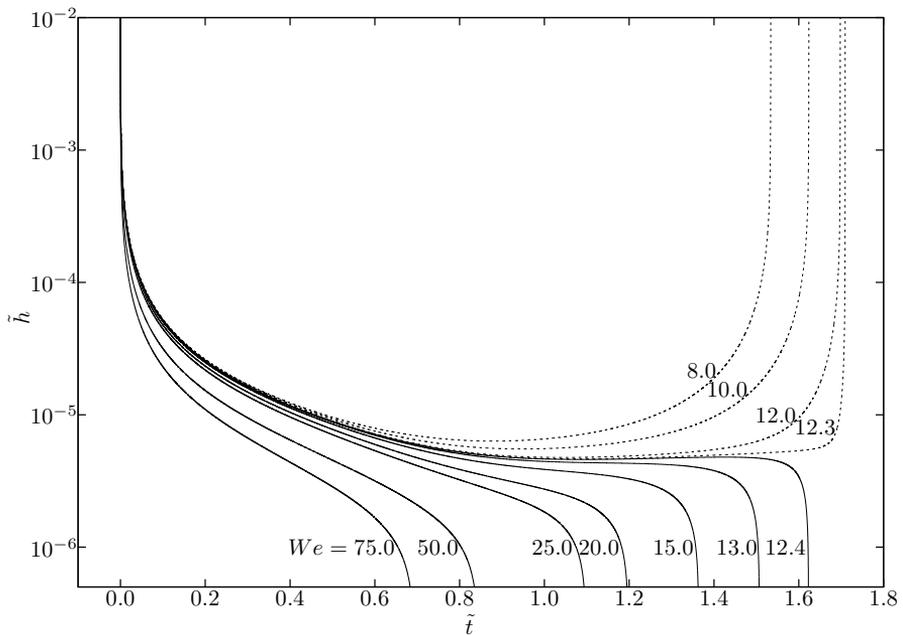
Fig. 3.4 shows the evolution of the film thickness with time for tetradecane droplets in air for various We , see Table 3.1 for physical properties and model parameters.

* Eq. (3.17b) is not obtained by matching the pressure at $r = 0$ as stated in [166], but by matching the interaction force.

† Although stated otherwise in [166], in a private correspondence the authors mentioned that $\delta = 1.3 \left(\frac{\mu_l / \sigma}{(\mu_l / \sigma)_{\text{tetradecane}}} \right)^{-1}$ was used in their work.



(a) Transition from coalescence to bouncing with increasing We .



(b) Transition from bouncing to coalescence with increasing We .

Figure 3.4: Evolution of the film thickness \tilde{h} with time \tilde{t} for the collision of tetradecane droplets in air for various initial We .

Table 3.1: Physical properties and model parameters for the collision of tetradecane droplets in atmospheric air.

Air density	ρ_g	1.225 kg m^{-3}
Air viscosity	μ_g	$1.827 \times 10^{-5} \text{ N s m}^{-2}$
Tetradecane density	ρ_l	762.0 kg m^{-3}
Tetradecane viscosity	μ_l	$2.128 \times 10^{-3} \text{ N s m}^{-2}$
Mean free path	λ_g	$69 \times 10^{-9} \text{ m}$
Surface tension	σ	$2.65 \times 10^{-2} \text{ N m}^{-1}$
Hamaker constant	A_H	$5.0 \times 10^{-20} \text{ J}$
Droplet radius	R_0	$168 \times 10^{-6} \text{ m}$
Empirical parameter	δ	1.837

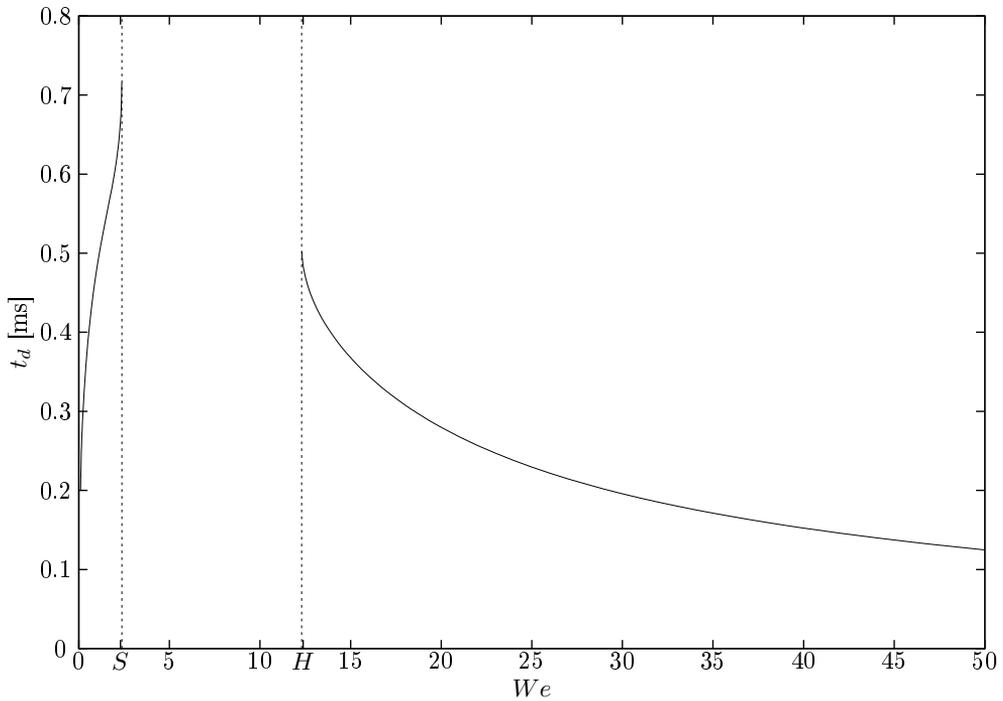


Figure 3.5: Evolution of the film drainage times for increasing We for $R_0 = 168\mu\text{m}$ droplets. The soft and hard transition boundaries are indicated by respectively S and H .

It demonstrates the non-monotonic coalesce-bounce-coalesce transition of collision outcomes: for $We < 2.4$ droplets coalesce, for $2.4 \leq We \leq 12.3$ droplets bounce, and for $We > 12.3$ droplets coalesce again. The experimentally obtained transition Weber numbers are $We_S = 2.3$ for soft collisions, and $We_H = 12.3$ for hard collisions [115], see Fig. 3.5. Although these values seem very accurate, it must be noted that

the empirical parameter δ and initial conditions strongly influence the location of these boundaries. Nevertheless, the model is able to predict the coalesce-bounce-coalesce transitions and shows the expected film drainage stages in time: rapid initial approach, slow film drainage, and rapid film rupture due to van der Waals forces.

Although the film drainage model is developed for head-on collisions of two equal sized droplets, it should be possible to extend the method to more general droplet collisions. In general collisions are oblique ($B \neq 0$) and the droplets have an unequal size ($R_1 \neq R_2$), see Fig. 3.6. Oblique collisions can be handled by decomposing the droplet velocity into a normal (V^n) and a tangential (V^t) part. Next, the oblique collision could be approximated by a head-on collision (where the collision We is based on the velocity V^n) with an additional centrifugal force (due to rotation by V^t). The centrifugal force should be included in Eq. (3.13d) via the force balance. Chesters and Hofman [23] showed that the equations governing film drainage for a collision between unequal sized droplets are the same as for a collision between equal sized droplets of equivalent radius, R_{eq} . However, by changing the droplet radii to R_{eq} the droplet momentum and energy balances are also affected. These balances should thus be properly matched as well. Since the emphasis of this work lies on head-on collisions of equal sized droplets, a physically sound extension of the method to more general droplet collisions is left for future research.

3.3.2 Breakup model

The breakup of a droplet into smaller fragments is usually caused by large velocity and pressure variations around or within the droplet. Such variations can originate from interaction of the continuous phase flow with the droplet, or by coalescence events between droplets. Pilch and Erdman [118] reviewed several breakup scenarios for an initially stable droplet in a flow, which are categorized by the initial We †. One can define a critical We , We_{cr} , below which no droplet breakup occurs. For We slightly larger than this threshold value, gentle breakup occurs in two or a limited number of child droplets. So-called vibrational breakup originates from oscillations at the natural frequency of the droplet, which can be enhanced by the surrounding flow field (e.g. turbulence). For We much higher than We_{cr} more complicated breakup mechanisms appear, which finally become chaotic. Another origin for breakup are coalescence events, where a surplus of kinetic and surface energy can lead to breakup (regime IV and V of Fig. 3.1). During the stretching of the just coalesced droplet a connecting ligament can be formed, which subsequently contracts to form one or more satellite droplets [121]. The physical mechanism causing ligament breakup is the so-called end-pinching [138, 139].

Without excessive grid refinement, the multiple marker CLSVOF method can not accurately capture extremely small droplet fragments. Therefore, only gentle breakup with as few satellite droplets as possible will be considered. In the present study no explicit physical breakup model is implemented, but breakup is accom-

† The continuous phase density is used to derive this We .

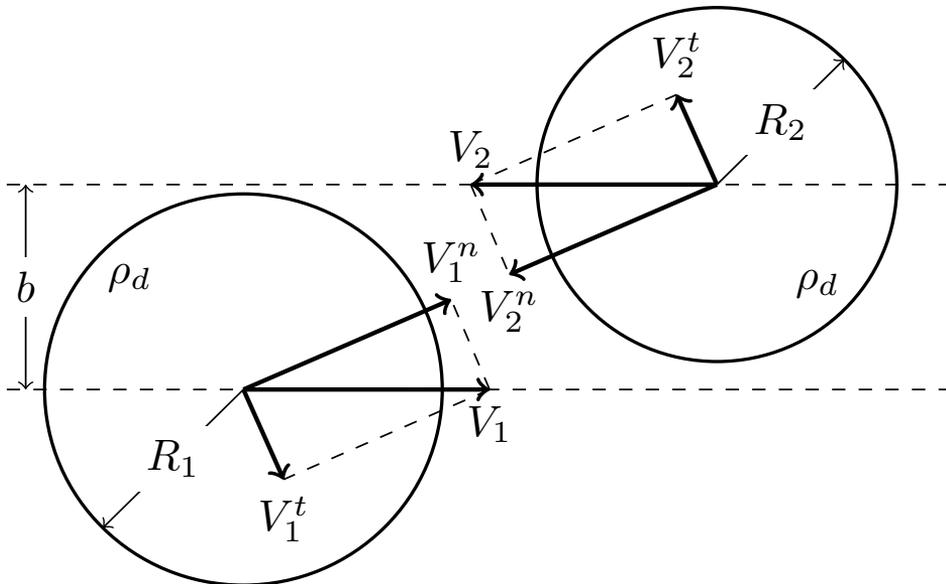


Figure 3.6: Nomenclature of the decomposition of a general binary droplet collision, where $B = b/(R_1 + R_2)$ is the impact parameter, V the droplet velocity, R the droplet radius, ρ_l the dispersed phase density, and σ the surface tension coefficient. The velocity V of each droplet is decomposed into a normal and a tangential part, denoted by V^n and V^t , respectively. The droplet radii are converted to an equivalent droplet radius, R_{eq} . The Weber number We is determined from the normal velocities and the equivalent droplet radius.

modated by means of a numerical algorithm. While the film drainage process prior to coalescence is a very slow process, the neck formation prior to breakup is a much faster process. It is therefore expected that the collision dynamics do not critically depend on the details of breakup. The breakup algorithm is detailed in the next section.

3.3.3 Numerical implementation

In this section the extension of the CLSVOF method with coalescence and breakup modules is discussed. The inclusion of both modules within the original interface update algorithm [75] is given in Alg. 6. The base of the algorithm is a linked list, which is available on each process of the parallel simulation. This linked list contains all droplets, but only the master of a droplet handles the computationally demanding tasks.

§ Since the governing equations require updated density, viscosity, and surface tension fields, masters should send certain data (e.g. $\phi^{n+\frac{1}{2}}$) which can provide these updated fields to their slaves [75].

Algorithm 6 Interface update subroutine at time level n

```

1: for all interfaces do
2:   Send  $\mathbf{u}^n$  from slaves to master
3: end for
4: for all interfaces do
5:   if master then
6:     Advect  $\psi^{n-\frac{1}{2}}$  with  $\mathbf{u}^n$  to  $\psi^{n+\frac{1}{2}}$ 
7:     Advect  $\phi^{n-\frac{1}{2}}$  with  $\mathbf{u}^n$  to  $\phi^*$ 
8:     Redistribute mass to minimize  $\int_{\Omega} \epsilon \, d\Omega$  (see Section 3.3.3)
9:     Detect breakup (see Alg. 8) and create separate markers
10:    Correct and reinitialize  $\phi^*$  to  $\phi^{n+\frac{1}{2}}$ 
11:   end if
12: end for
13: Update breakup: remove parent and add children to global marker list
14: Detect collisions & coalesce markers
15: for all interfaces do
16:   Send dataS from master to slaves
17: end for

```

In Step 9 a breakup detection algorithm (see Section 3.3.3), is used to split one parent marker in multiple children. By splitting the parent into multiple children before the reinitialization, during the reinitialization only LS and VOF values of one specific child are used. Note that at this point only the master of the parent that has been split is aware of this breakup event. After all interfaces have been advected and reinitialized, in Step 13 newly created children are added to the global marker list to inform all other processes.

Next, in Step 14 collisions between interfaces are detected. The moment of first contact is stored, from which the contact time of two interfaces can be computed later on. As long as a collision remains active it can result in coalescence. A more detailed discussion of the coalescence module is given in Section 3.3.3.

Coalescence module

In this work a novel approach is presented to handle coalescence, where each collision is treated as unique. The collision outcome is determined by comparing the numerically measured contact time of two colliding droplets with the computed film drainage time from the model of Section 3.3.1. The contact time is related to the question when a collision starts, which definition is somewhat arbitrary. However, the exact moment of the start of a collision is of minor importance for the remaining drainage time, since initially the film drains very fast compared to the later stage of drainage [126]. The volume and velocity vectors of the droplets, which are obtained from the multiple marker CLSVOF model, are used as input variables for the film drainage model. Numerically this approach is handled by a coalescence module, which extends the method as presented in Kwakkel et al. [75].

The coalescence module is implemented as given in Alg. 7. In Step 3, on all slaves

Algorithm 7 Coalescence module

```

1: for all interfaces do
2:   if slave then
3:     Determine domain overlap with other slaves on processor
4:     for all cells in overlapping domain do
5:       Determine if collision is active
6:       if collision is active then
7:         Initialize/update contact time
8:       else
9:         Remove collision from collision list
10:      end if
11:    end for
12:  end if
13: end for
14: for all interfaces do
15:   Send collisions from slaves to master
16: end for
17: for all interfaces do
18:   if master then
19:     Remove duplicate collisions from collision list
20:   end if
21: end for
22: Synchronize collision data (volume, velocity vector) between masters
23: Compute drainage time & check for coalescence on masters

```

of a marker the domain overlap between other interfaces is determined. Only within this overlap domain the LS and VOF values of the two interfaces are investigated, which reduces unnecessary overhead. If in a cell $VOF < 1$ and the sum of the absolute values of the two LS functions is $< h/2$ (where h is the grid cell size), then the collision is initiated or updated. At initiation, the collision and its starting time are stored in a collision list (linked list). Each time step this collision list is updated, where the contact time of active collisions is increased with the time step, and inactive collisions are removed. Every time step the collision list is synchronized between all processors, where duplicate collisions are removed. These duplicates are present since collisions are detected at slave level.

When a collision is initiated, the film drainage time is computed via the film drainage model. The CLSVOF method provides the volume and velocity vector (by a VOF weighted average of velocities interpolated to cell centers) of each droplet. From the volume of the droplet the initial radius R_0 is obtained by assuming that its shape is a perfect sphere. From the velocity vectors a We can be computed, but also the amount of offset B between the droplets. All these parameters are passed on to the film drainage model, from which the drainage time is computed. If during

a collision the contact time exceeds the predicted film drainage time, then the two marker functions are merged by means of numerical coalescence.

Breakup module

When a droplet breaks up into two or more smaller droplets, the newly formed droplets need to have their own marker functions in order to prevent future numerical coalescence when they would encounter each other again. First the breakup of a droplet has to be detected, which is done by searching for unconnected features within a single marker function. This breakup detection algorithm is given by Alg. 8 and graphically illustrated in Fig. 3.7. The algorithm shows similarities with the algorithm presented by Herrmann [57], although here recursivity is used.

Algorithm 8 Breakup detection algorithm

```

1: Reset all  $\text{tag}_c$  to 0
2: Reset level to 0
3: for all cells  $c$  do
4:   if  $\phi_c > 0$  and  $\text{tag}_c = 0$  then
5:     level = level + 1
6:     procedure RECURSE(cell  $c$ )
7:       Set  $\text{tag}_c = \text{level}$ 
8:       for all 26 direct neighbors do
9:         if  $\phi_c > 0$  and  $\text{tag}_c = 0$  then
10:          Call RECURSE(cell  $c$ )
11:        else if  $\text{tag}_c = 0$  then
12:          Set  $\text{tag}_c = \text{level}$ 
13:        else
14:          Set  $\text{tag}_c = -1$ 
15:        end if
16:      end for
17:    end procedure
18:   end if
19: end for

```

This algorithm recursively tags all connected cells with positive ϕ values with a certain level integer. The boundary of this feature is also tagged with the same level integer. If such a boundary cell is already tagged by another (unconnected) feature, then this cell is retagged with -1 to indicate that it is a shared cell. After this algorithm each unconnected feature has its own level, which can be used as a mask to split the parent droplet into multiple children. The ψ fraction of shared cells is equally divided between children.

When a breakup event is detected, only the master process of the parent is aware of this event. Since each process needs to maintain a synchronized linked list of

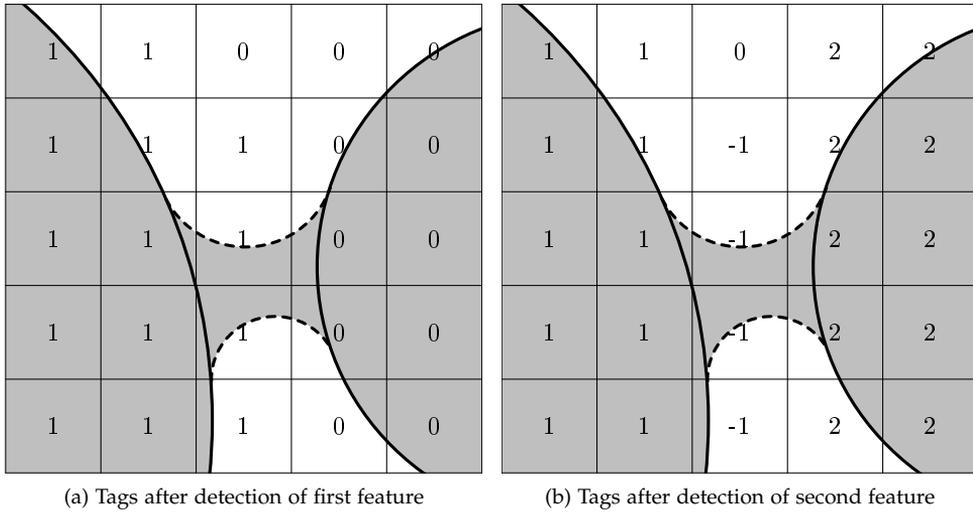


Figure 3.7: Two-dimensional example of breakup detection.

interfaces, a notification is sent to the remaining processes. These processes remove the parent and add the newly created children to their linked list.

Mass redistribution

During coalescence and breakup events small fragments (continuous phase inclusions, small dispersed phase droplets) can be formed. Although the formation of such small fragments can be physical, due to lack of spatial resolution they can not be accurately captured. Moreover, within the CLSVOF method such small fragments easily lose mass, which should be prevented. Such mass errors originate from truncation of volume fractions by two effects: to correct for undershoots ($\psi < 0$) and overshoots ($\psi > 1$), and to correct for cells with $|\phi| > h$ (no interface present), but a non-zero ψ value (still a volume present). The first effect is a result of numerical errors due to a spatially varying vorticity, by which some volume is fluxed more than once or not fluxed at all [124] during the advection of ψ by Eq. (3.10). Mass errors due to these under- and overshoot are small in general, typically $\mathcal{O}(10^{-4})$. The second effect avoids generation of spurious volume fractions, so-called flotsam and jetsam. This is necessary in order to guarantee that the minimization problem to linearly reconstruct the interface always has a solution [143]. However, by bluntly truncating volume, mass errors can become relatively large. After the CLSVOF advection, see Alg. 6, ψ is mass-conserving but can contain unwanted values. To prevent mass loss/gain, a similar mass redistribution algorithm as presented by [117] is used. The first step is to split the updated ψ as

$$\psi = \hat{\psi} + \epsilon \quad (3.18)$$

where $\hat{\psi}$ contains corrected values, and ϵ the occurring mass errors. Without redistribution, the total mass loss/gain would be equal to $\int_{\Omega} \epsilon \, d\Omega$. However, by iteratively redistributing ϵ towards the nearest interface cells with $0 < \hat{\psi} < 1$ and adding it back to ψ , the total mass loss/gain is reduced or even prevented at all. By redistributing only towards the nearest interface, the effect on ϕ is minimized. To find the nearest interface a mapping is used, which is given by Alg. 9 and graphically illustrated in Fig. 3.8.

Algorithm 9 Mapping algorithm

```

1: Reset map array to -1
2: Reset level integer to 0
3: for all cells  $c(i, j, k)$  which contain the interface do
4:   Set  $\text{map}(i, j, k) = \text{level}$ 
5: end for
6: procedure RECURSE(level)
7:   Set  $\text{newLevel} = \text{level} + 1$  and  $\text{next} = \text{false}$ 
8:   for all cells with  $\text{map}(i, j, k) = \text{level}$  do
9:     for all neighbors of  $c(i, j, k)$  with  $\text{map} = -1$  do
10:      Set neighbor  $\text{map} = \text{newLevel}$ 
11:      Set  $\text{next} = \text{true}$ 
12:     end for
13:   end for
14:   if  $\text{next}$  then
15:     call RECURSE(newLevel)
16:   end if
17: end procedure

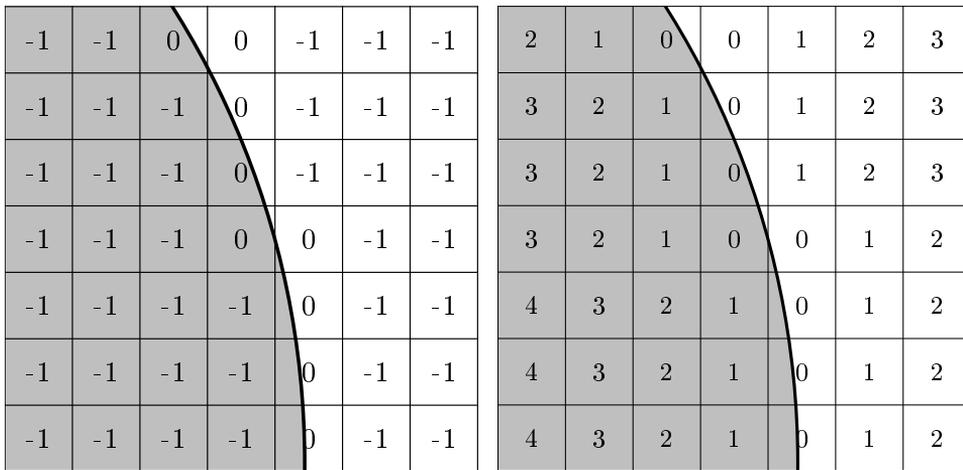
```

Finally, the ϵ distribution is conservatively distributed from cells with high tag values to low tag values. If a cell has multiple neighboring cells with equal tag values, the local ϵ is equally distributed over these neighbors. A final distribution is usually obtained after a few iterations, and the total mass loss/gain depends on the error in the divergence of the velocity field, which is $\mathcal{O}(10^{-11})$.

3.4 NUMERICAL RESULTS

To validate the new method, numerical simulations of head-on collisions between equal sized droplets are performed and compared with experimental data. Although symmetry conditions allow simulation of only a fraction of the domain, here the full domain with two droplets is simulated. The method can therefore also be applied to oblique and unequal sized droplet collisions, without the use of symmetry boundary conditions to enforce droplet coalescence [64, 79].

Unless indicated otherwise, the droplet diameter D is resolved by 24 grid points of dimension h , so $D/h = 24$. A uniform Cartesian grid without any refinement in



(a) Mapping of all cells which contain the interface (Alg. 9, after completing Step 5).

(b) Mapping after completing Alg. 9.

Figure 3.8: Two-dimensional domain mapping example.

the film region is used. The computational domain has dimensions $4D \times 5D \times 4D$ with the largest dimension in the direction of the collision, and free slip boundary conditions are used on all six sides of the domain. Initially the distance between the droplet centers is $1.4D$. An initial velocity field is used, which imposes a uniform velocity of V_0 only on the droplets in opposite directions without any driving force. These boundary conditions were also used by Pan and Suga [116] for their investigation of binary droplet collisions.

To validate the multiple marker CLSVOF method, in Section 3.4.1 collision dynamics of binary (near) head-on collisions is discussed. The instant of coalescence is determined from the experimentally observed film drainage times, which is a common approach for binary droplet collisions [105, 107, 115]. Droplet collisions in the four distinct collision regimes for head-on droplet collisions, as shown in Fig. 3.1, are compared with available numerical and experimental results. Next, in Section 3.4.2 the effect of the prescribed film drainage time on collision dynamics is investigated. Finally, in Section 3.4.3 the effect of grid resolution on the collision dynamics is studied.

3.4.1 Collision dynamics based on experimentally observed drainage times

Fig. 3.1 shows four distinct regimes for equal sized head-on ($B = 0$) droplet collisions. In this section collisions within each of these four regimes are simulated with the multiple marker CLSVOF method, where the coalescence instant is prescribed by experimental observations. The simulation results will demonstrate the ability

of the CLSVOF method to capture droplet dynamics during both bouncing and coalescing collisions.

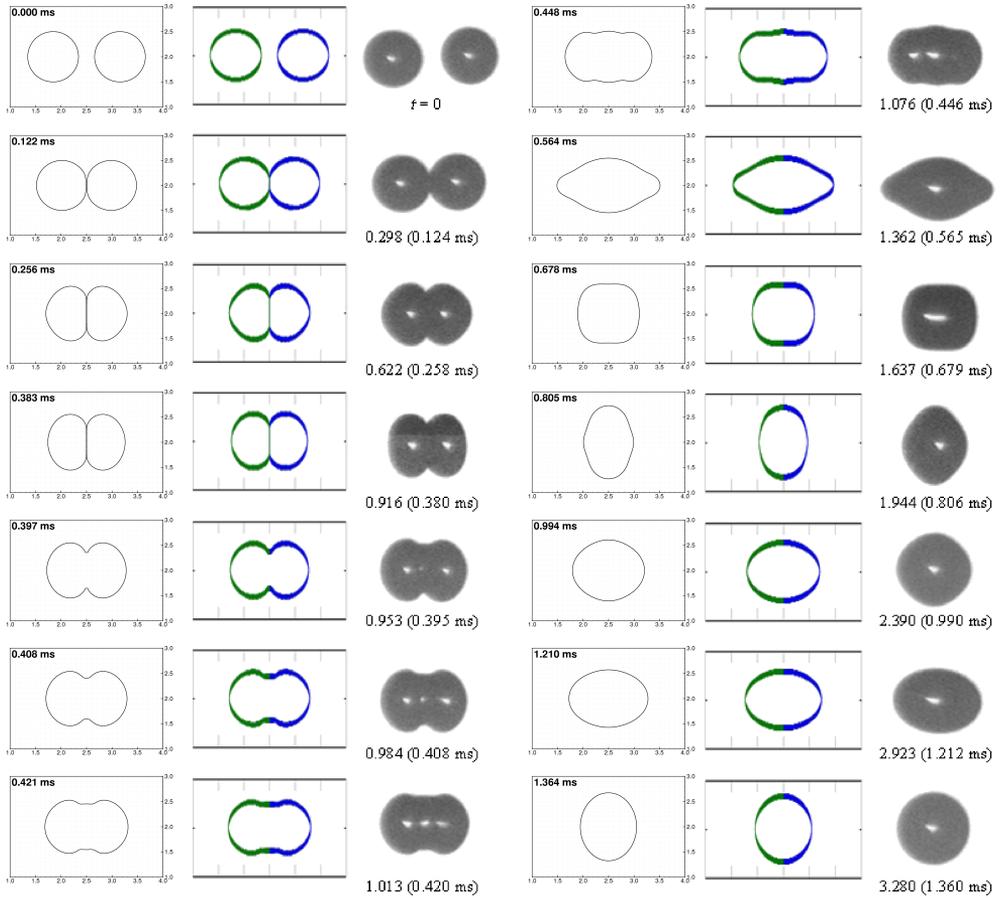


Figure 3.9: Merging collision sequence (in regime I) near the soft transition boundary. Results from left to right are: computation with CLSVOF method (present study), computation with front-tracking method [115], and experimental results of [115]. Conditions: tetradecane in 1 atm. air, $R = 107.2 \mu\text{m}$, $V_0 = 0.305 \text{ m/s}$, $We = 2.3$, and $B = 0$. The physical film drainage time is approximately $t_d = 0.270 \text{ ms}$.

Regime I, which applies to gentle collisions, always results in coalescence. Fig. 3.9 shows such a collision for tetradecane droplets in 1 atm. air, with $R = 107.2 \mu\text{m}$, $V_0 = 0.305 \text{ m/s}$, and $We = 2.3$. Pan et al. [115] experimentally observed contact around $t = 0.124 \text{ ms}$, and coalescence at $t = 0.388 \text{ ms}$. This results in an experimentally observed film drainage time of about $t_d = 0.264 \text{ ms}$. The numerical results of the CLSVOF method with this prescribed film drainage time are compared with numerical (front-tracking) and experimental results of [115]. Both numerical results agree well with

the experimental results, which shows the accuracy of these interface resolving methods.

Regime II always results in bouncing, since the contact time in this regime is too short to allow full film drainage. Fig. 3.10 shows such a collision for tetradecane droplets in 1 atm. air, with $R = 167.6 \mu\text{m}$, $V_0 = 0.492 \text{ m/s}$, and $We = 9.33$. Again, both numerical results agree well with the experimental results.

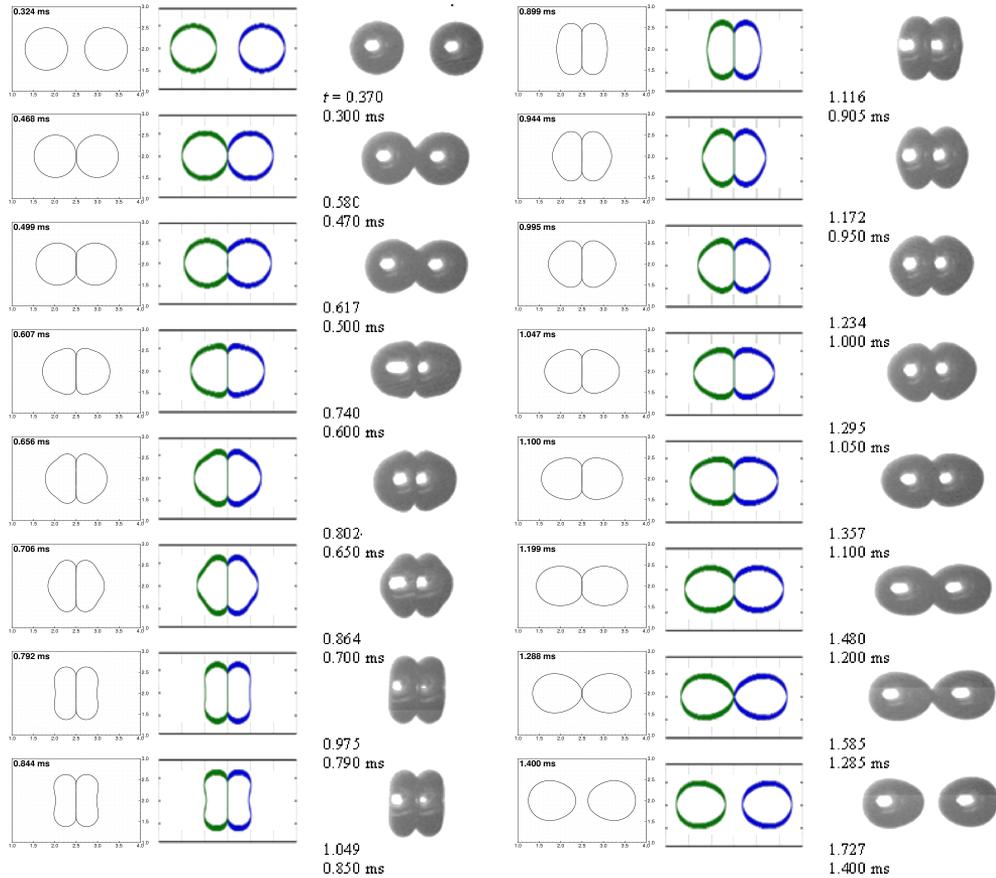


Figure 3.10: Bouncing collision sequence (in regime II) near the hard transition boundary. Results from left to right are: computation with CLSVOF method (present study), computation with front-tracking method [115], and experimental results of [115]. Conditions: tetradecane in 1 atm. air, $R = 167.6 \mu\text{m}$, $V_0 = 0.492 \text{ m/s}$, $We = 9.33$, and $B = 0$.

Regime III results in coalescence, but the collision is not gentle as is the case in regime I. Due to the high kinetic energy the deformation of the droplets is substantial, however the film has enough time to reach a critical thickness at which rupture occurs. Fig. 3.11 shows such a collision for tetradecane droplets in 1 atm. air, with $R = 169.7 \mu\text{m}$, $V_0 = 0.591 \text{ m/s}$, and $We = 13.63$. At $t = 0.500 \text{ ms}$ the numerical

cross sections show dimples at both sides of the droplet, which are hidden from the visual field in the experiment. Furthermore, the small deviations between the CLSVOF method and the front-tracking method after $t = 0.500$ ms are probably a result of a slightly delayed coalescence in the CLSVOF method. However, overall the CLSVOF method is in agreement with both the numerical and experimental results of [115].

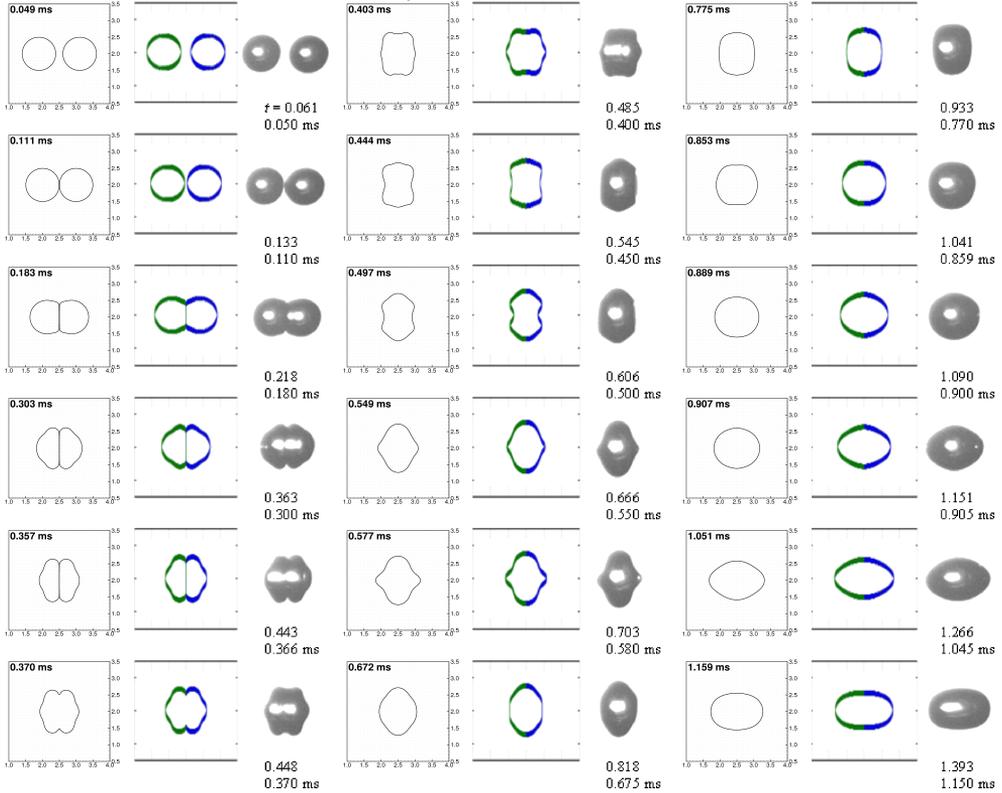


Figure 3.11: Merging collision sequence (in regime III) near the hard transition boundary. Results from left to right are: computation with CLSVOF method (present study), computation with front-tracking method [115], and experimental results of [115]. Conditions: tetradecane in 1 atm. air, $R = 169.7 \mu\text{m}$, $V_0 = 0.591 \text{ m/s}$, $We = 13.63$, and $B = 0$. The physical film drainage time is approximately $t_d = 0.246 \text{ ms}$.

In regime IV the kinetic energy is sufficient to overcome the surface energy, and causes separation of the just coalesced droplet into a number of smaller droplets. For tetradecane droplets in 1 atm. nitrogen, with $R = 168 \mu\text{m}$, $We = 61.4$, and $B = 0.06$, such a collision is shown in Figs. 3.12, 3.13 and 3.14, where the grid resolutions are $D/h = 24$, $D/h = 48$ and $D/h = 60$, respectively. To compare with available experimental results, the collision is slightly oblique, the continuous phase is nitrogen ($\rho_g = 1.138 \text{ kg m}^{-3}$, $\mu_g = 1.787 \text{ N s m}^{-2}$, $\sigma = 0.026 \text{ N m}^{-1}$) instead of air,

and a tetradecane density of $\rho_d = 758 \text{ kg m}^{-3}$ is used. Although the film drainage time can not be determined from the experimental results, the high We justifies the use of a zero film drainage time. For the coalescence of the satellite droplet with the torus shape around $t = 0.68 \text{ ms}$ also a zero film drainage time is used (see Fig. 3.15, which is just prior to this event).

The numerical results largely agree with the experimental results, although all simulations show premature separation. After $t = 0.720 \text{ ms}$ the numerical results start to deviate from each other. The origin of this difference is the less accurate curvature estimation between $t = 0.350$ and $t = 0.720 \text{ ms}$ for $D/h = 24$, which is caused by the limited number of cells in strongly deformed regions. This slight error in curvature estimation has a direct consequence for the acting surface tension force, and over time the surface shape for $D/h = 24$ is severely influenced. This effect is visible in Fig. 3.15, where the droplet shapes around $t = 0.61 \text{ ms}$ are shown for four subsequent grid resolutions. For $D/h = 48$ and $D/h = 60$ the expected circular shapes are obtained, but $D/h = 24$ and $D/h = 36$ show stronger grid dependency. Due to this grid resolution effect the available Gibbs free energy for finer grids is slightly larger, resulting in a more elongated droplet prior to separation.

Since no physical breakup criteria are used, the exact instant of separation depends somewhat on the grid resolution. A finer grid can capture the thin neck longer, but will eventually always lead to so-called numerical breakup. A postponed breakup will probably result in a larger satellite droplet, as shown by Pan and Suga [116] (oblique collision) and Li and Fritsching [79] (head-on collision). Fig. 3.16 shows the creation of new local marker functions when unconnected features are detected by Alg. 8. These local markers are equally divided among the available processes in the parallelization, as described by Kwakkel et al. [75].

3.4.2 *Effect of film drainage time on collision dynamics*

In the previous section experimentally observed film drainage times were used in the numerical simulations. The results show that if these film drainage times are used within the coalescence model, the CLSVOF method can well predict the droplet dynamics. However, typically these drainage times are unknown for general droplet collisions, which makes modeling of these drainage times inevitable. The film drainage model of Zhang and Law [166] as presented in Section 3.3.1 predicts both the outcome of a collision as well as the film drainage time prior to coalescence.

To study the importance of the film drainage time on the droplet dynamics, the gentle collision as shown in Fig. 3.9 is solved for varying film drainage times. Fig. 3.17 shows, from left to right, the droplet dynamics for $t_d = 0.000 \text{ ms}$ (corresponding to the case of the single marker CLSVOF method), $t_d = 0.235 \text{ ms}$ (early coalescence), $t_d = 0.270 \text{ ms}$ (the experimentally observed film drainage time, as used in Fig. 3.9), $t_d = 0.340 \text{ ms}$ (delayed coalescence, predicted by the film drainage model), and $t_d > t_c$ (bouncing, as would occur for the multiple marker CLSVOF method of Coyajee and Boersma [27]). As expected, clear differences between droplet dynamics for varying film drainage times appear. For $t_d = 0.000 \text{ ms}$ and $t_d = 0.235 \text{ ms}$ the droplets coalesce before the instant of maximum deformation,

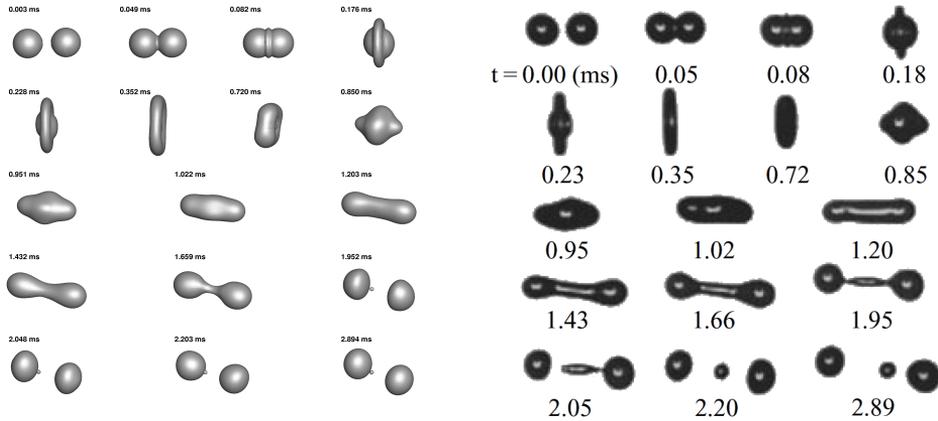


Figure 3.12: Near head-on separating collision sequence (in regime IV) for tetradecane droplets in nitrogen. Computation with the CLSVOF method with $D/h = 24$ (left) compared to experimental results of Qian and Law [121] (right). Conditions: tetradecane in 1 atm. nitrogen, $R = 168 \mu\text{m}$, $V_0 = 1.260$, $We = 61.4$, and $B = 0.06$. The physical film drainage time is assumed to be zero.

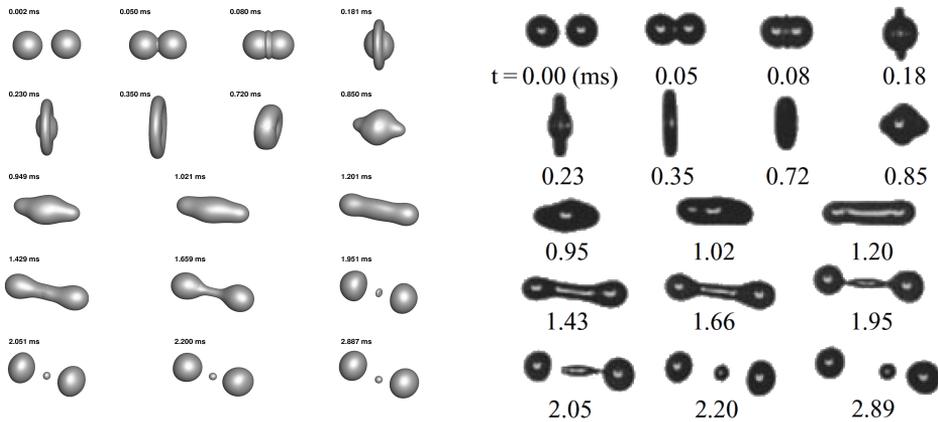


Figure 3.13: Near head-on separating collision sequence (in regime IV) for tetradecane droplets in nitrogen. Computation with the CLSVOF method with $D/h = 48$ (left) compared to experimental results of Qian and Law [121] (right). Conditions: tetradecane in 1 atm. nitrogen, $R = 168 \mu\text{m}$, $V_0 = 1.260$, $We = 61.4$, and $B = 0.06$. The physical film drainage time is assumed to be zero.

thereby first flattening the droplets followed by elongation in the horizontal direction. For $t_d = 0.270$ ms and $t_d = 0.340$ ms the droplets coalesce after the instant

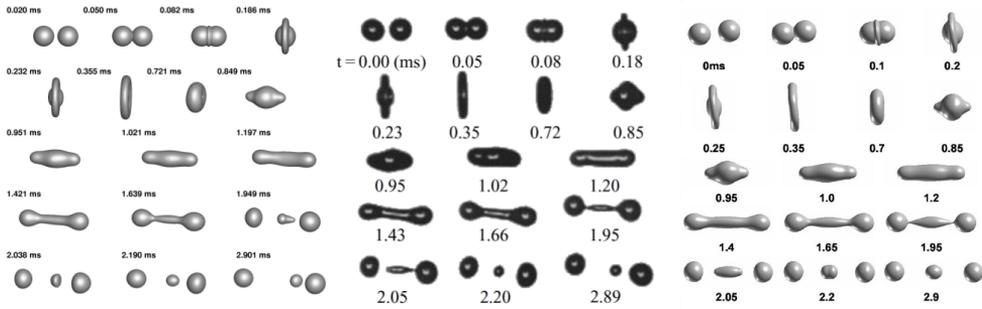


Figure 3.14: Near head-on separating collision sequence (in regime IV) for tetradecane droplets in nitrogen. Computation with the CLSVOF method with $D/h = 60$ (left), compared to experimental results of Qian and Law [121] (middle) and the computational results with the LS method with $D/h = 60$ of Pan and Suga [116] (right). Conditions: tetradecane in 1 atm. nitrogen, $R = 168 \mu\text{m}$, $V_0 = 1.260$, $We = 61.4$, and $B = 0.06$. The physical film drainage time is assumed to be zero.

of maximum deformation, which can even result in separation if the droplets coalesce near the end of the contact period. If the film drainage time t_d is longer than the contact time t_c , coalescence will be prevented and the droplets will always bounce. The exact instant of coalescence has therefore a large influence on the subsequent droplet dynamics, and accurate prediction is especially important near the coalescence-separation boundary in the We - B domain (boundary between collision regime III and IV in Fig. 3.1).

The model of Section 3.3.1 predicts a dimensionless film drainage time for $We = 2.3$ of $\tilde{t}_d = 0.9674$. Since the time scale for droplets with $R = 107.2 \mu\text{m}$ is $t = 0.3510 \text{ ms}$, the dimensional film drainage time becomes $t_d = 0.340 \text{ ms}$. This value is nearly 26 % higher than the experimentally observed value of 0.270 ms. For $We = 13.63$ the model predicts a dimensionless film drainage time of $\tilde{t}_d = 1.4474$. For droplets with $R = 169.7 \mu\text{m}$, which have a time scale of 0.2872, this leads to a dimensional film drainage time of $t_d = 0.416 \text{ ms}$. This value is also higher than the experimentally observed value of 0.246 ms. Though the order of magnitude of the predicted film drainage times is correct, a better quantitative agreement is desired since the collision dynamics appears to be sensitive to the value of the film drainage time. This sensitivity underlines the importance for accurate prediction of the film drainage time and the need for improvement of the currently used film drainage model.

3.4.3 Effect of grid resolution on collision dynamics

Except for the collision in regime IV, all previous results are obtained with $D/h = 24$. If the interface deformation is not too severe, $D/h = 24$ is usually sufficient to adequately capture the details of the interface dynamics [27]. However, to verify

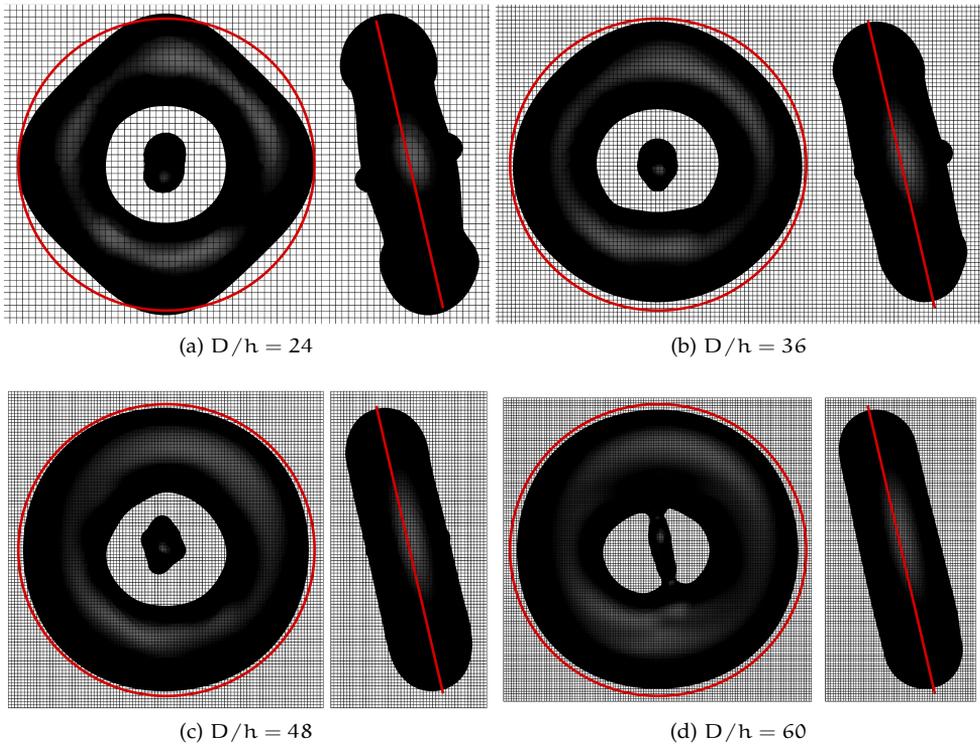


Figure 3.15: Grid resolution effect around $t = 0.61$ ms for four different resolutions. The red circle and slanted line have in each figure the same position and size.

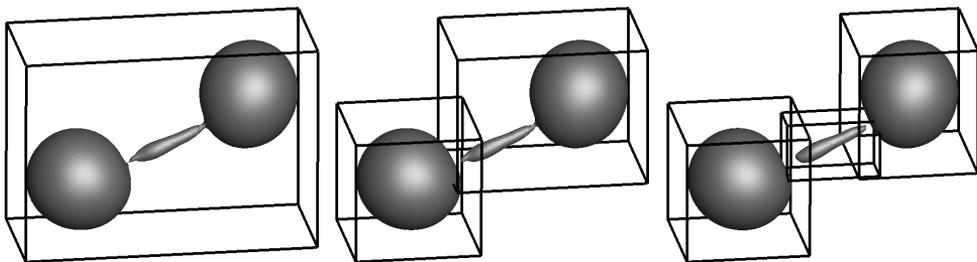


Figure 3.16: Local markers around the instant of droplet breakup (near head-on separation case with $We = 61.4$). Just before the neck breaks, only one local marker is used (left). Next, when unconnected structures within this marker domain are detected, each structure obtains its own local marker function (middle and right).

the influence of grid resolution on the coalescence (modeling), also simulations with $D/h = 32$ and $D/h = 48$ have been performed. Fig. 3.18 shows a comparison for the collision as shown in Fig. 3.11, where the film drainage time is fixed to 0.258 ms. The largest visible difference between these three grid resolutions is the delay of film rupture for finer grids. This is a direct result of the grid dependency of the start of the contact time. If the interfaces of two marker functions share the same computational cell, contact is detected and the contact timer is started. However, for finer grids this instant is slightly delayed. The difference between the start of the contact timer for $D/h = 24$ (coalescence at 0.381 ms) and $D/h = 48$ (coalescence at 0.398 ms) is 0.017 ms. Although the effect of this small delay on the further development of the droplet dynamics is limited, a grid independent start of the contact time is preferred. This can be achieved by starting the contact timer when the distance between the droplet centers is smaller than the sum of the two radii. Except for the effect on the start time of contact, the collision dynamics appear almost insensitive to the spatial grid resolution.

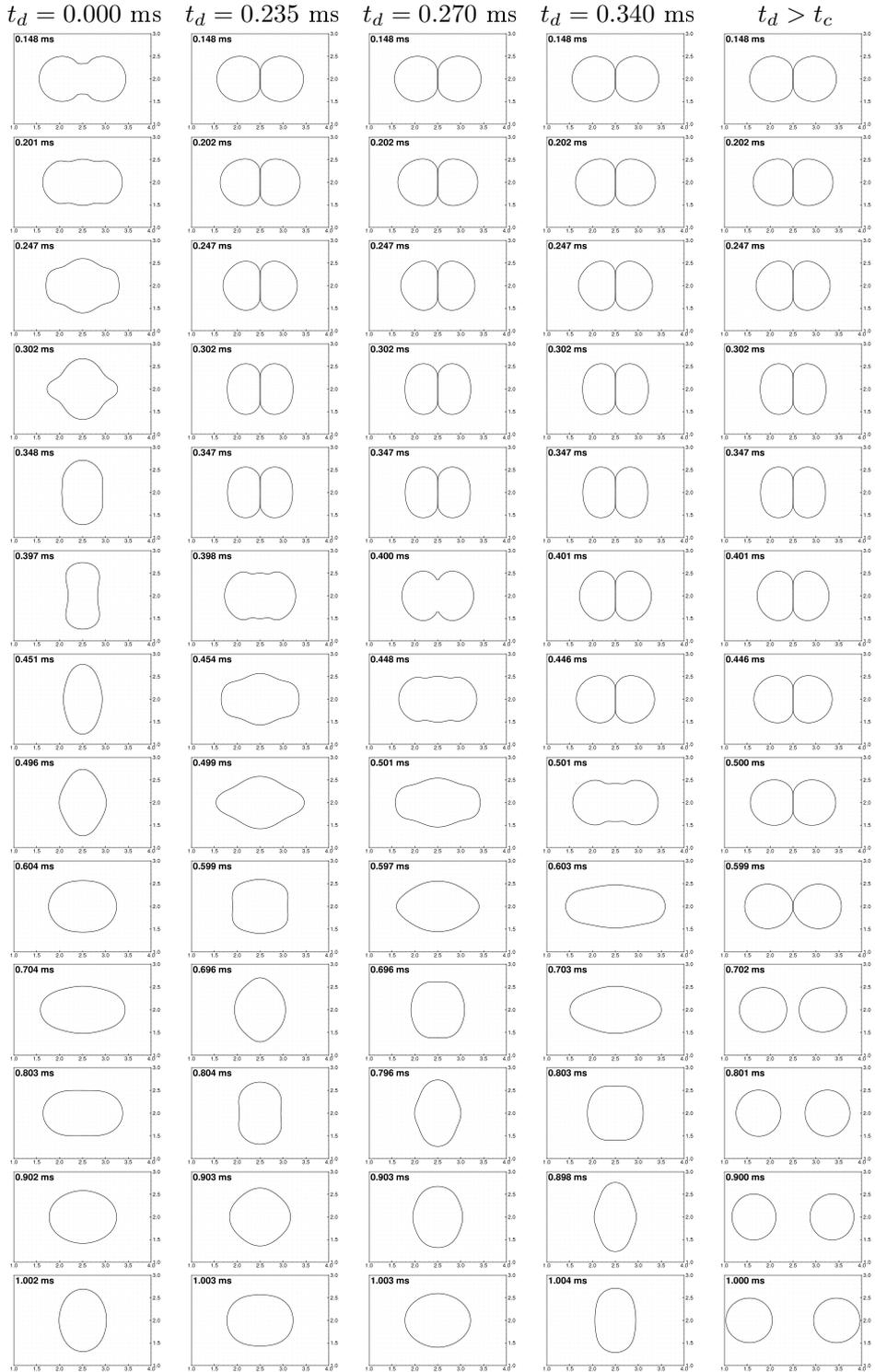
3.5 CONCLUSIONS AND DISCUSSION

The simulations of (near) head-on collisions of two equal sized droplets presented in this work are in good agreement with previous numerical investigations of collisions [64, 79, 105, 107, 115, 116]. Here we presented a novel approach to determine the outcome of a binary droplet collision. Within this approach a coalescence model determines if and when two colliding droplets should coalesce. Although this idea was already suggested by the group of Tryggvason [107, 155], the present work is the first in which it is applied within an interface resolving method.

The coalescence model is based on a computationally efficient film drainage model [166], from which the film drainage time is determined. The coalescence model is incorporated within a multiple marker CLSVOF method [27], which is capable of simulating $\mathcal{O}(1000)$ dispersed droplets [75]. Although such an incorporation within an interface resolving method is laborious, it is considered inevitable for study of dense droplet-laden flows where droplets will frequently coalesce/breakup. Moreover, since the implementation of the coalescence model in the

Figure 3.17 (*facing page*): Collision sequences near the soft transition boundary to demonstrate the effect of the drainage time on the dynamics and outcome of the collision. The far left column shows immediate (numerical) coalescence, the middle column is close to the experimentally observed [115] collision outcome, and the far right column shows droplet bouncing (film drainage time $>$ contact time). Conditions: tetradecane in 1 atm. air, $R = 107.2 \mu\text{m}$, $V_0 = 0.305 \text{ m/s}$, $We = 2.3$, and $B = 0$.

3.5 CONCLUSIONS AND DISCUSSION



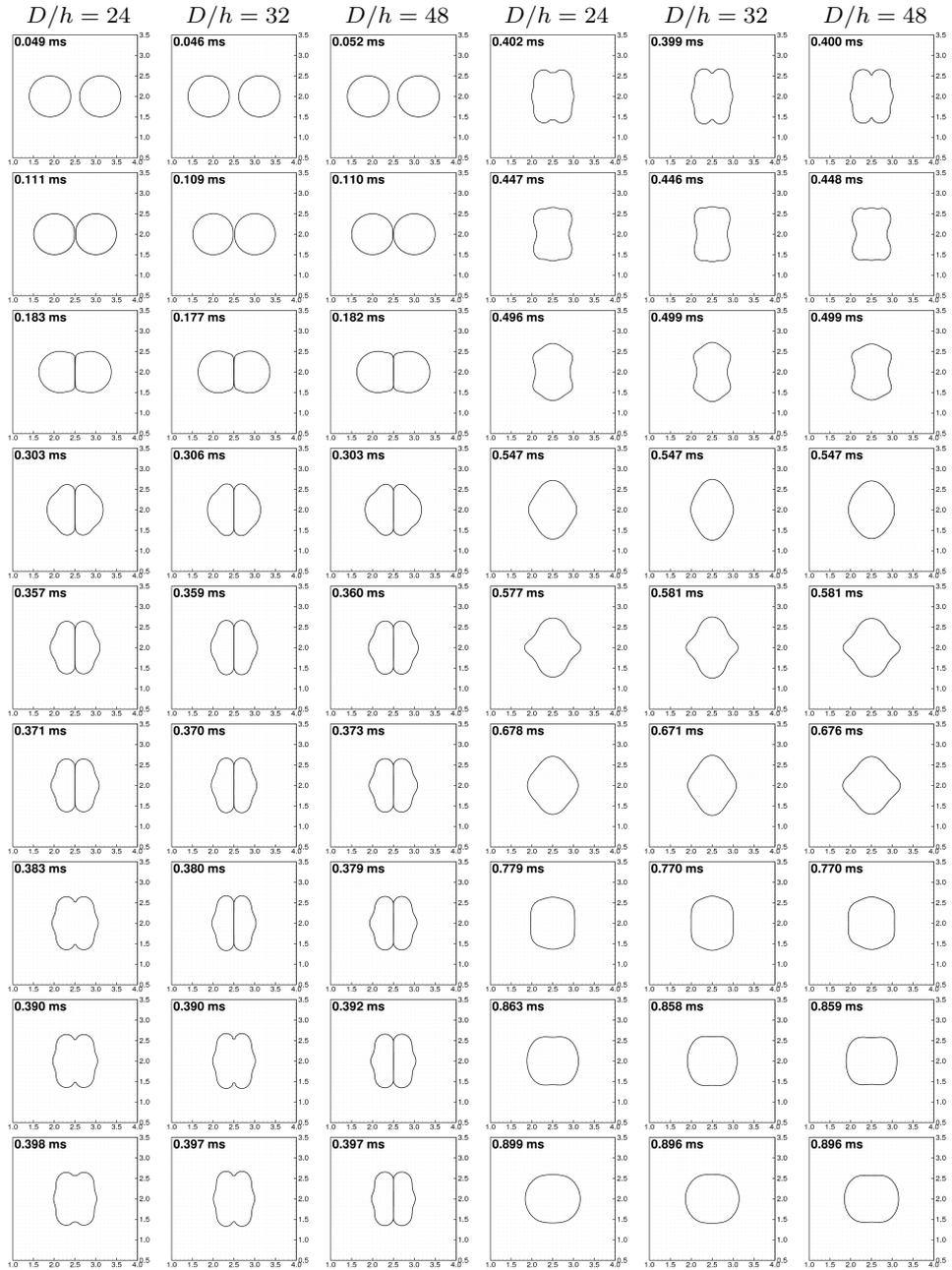
CLSVOF method is generic, incorporation of coalescence models other than the currently used film drainage model is straightforward.

To validate the extension of the CLSVOF method with the coalescence model, first experimentally observed film drainage times have been used. The results of these binary droplet collisions show that, if a correct film drainage time is available, the CLSVOF method can reproduce existing experimental and numerical results. This implies that the droplets are rather insensitive to the details of the simulated flow within the film, since these effects are not fully resolved. If the amount of deformation is large, the computed interface curvature becomes less accurate. These errors in curvature directly influence surface tension forces, and thereby over time the interface shape. Curvature estimation can be improved by adding more grid cells, but also by using more sophisticated algorithms to compute the curvature from LS and/or VOF functions [81, 93, 95], especially in off-diagonal directions. Since the number of grid cells over the diameter of small child droplets is usually insufficient, one could choose to represent these small child droplets by undeformable spherical point droplets.

Next, to study the influence of the film drainage time on the collision dynamics, several film drainage times for a specific collision have been analyzed. As expected, the results show that the collision dynamics strongly depends on the film drainage time. Therefore the coalescence model should provide accurate film drainage times to ensure realistic collision dynamics. Though the order of magnitude of the modeled film drainage times is correct, the quantitative prediction is not accurate enough. Possible causes for this deviation are the empirical parameter δ (which has no physical meaning), the truncated-sphere approximation of the droplets, and the assumption of axisymmetric film drainage (some experiments have shown asymmetric drainage [111]). However, due to the complexity of the film drainage problem it may not be straightforward to develop more accurate film drainage models with similar computational efficiency. The effect of grid resolution on the coalescence process is limited. Only the instant of first contact is slightly grid dependent, but this effect can easily be circumvented. The detection of breakup and the corresponding creation of new marker functions is essential to ensure physical coalescence events at later stages of the simulation. Since droplet breakup is a fast process, the effect of numerical breakup on the collision dynamics is negligible. Therefore, these aspects should be considered less important than the accuracy of the film drainage model, which has by far more influence on the collision dynamics.

Figure 3.18 (*facing page*): Collision sequences near the hard transition boundary to demonstrate the effect of grid resolution on the dynamics of the collision. The left column uses the default $D/h = 24$, the middle column $D/h = 32$, and the right column $D/h = 48$. Conditions: tetradecane in 1 atm. air, $R = 169.7 \mu\text{m}$, $V_0 = 0.591 \text{ m/s}$, $We = 13.63$, and the drainage time is fixed to 0.258 ms.

3.5 CONCLUSIONS AND DISCUSSION



Besides improving the film drainage model for head-on equal sized droplet collisions, extension to oblique unequal sized droplet collisions is desired. For oblique collisions a decomposition of the velocity into a normal and a tangential part, and for unequal sized collisions the equivalent droplet diameter [23] can be of use. To validate such models, more experimental results with various initial parameters and accurate measurements of film drainage times should be performed.

TURBULENT BUBBLY DOWNFLOW IN A VERTICAL CHANNEL

A person who never made a mistake never tried anything new

— Albert Einstein

4.1 INTRODUCTION

Turbulent two-phase bubbly flows are frequently encountered in natural and industrial processes. Examples of such processes are inclusion of air bubbles at the ocean surface (which contributes to the absorption of carbon dioxide by the ocean), heat transfer through boiling in power plants, and oxygen aeration/ozonation systems for water treatment. Their efficiency strongly depends on parameters like the (relative) bubble velocity, the bubble interfacial area, and the bubble shape, which directly influence momentum, heat and mass transfer, and chemical reactions. Fig. 4.1 illustrates such flows by showing two photos of rising bubbles in water.

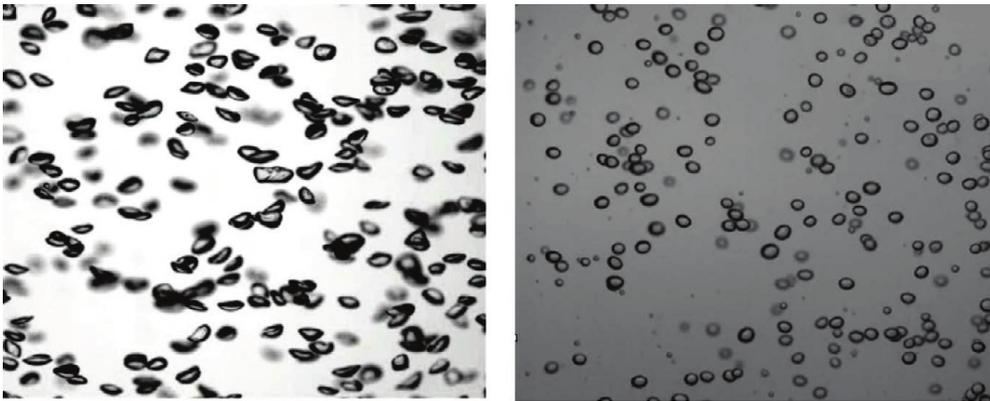


Figure 4.1: Left panel: rising bubbles of diameter ~ 5 mm in tap water. Right panel: bubbles of diameter ~ 3 mm in surfactant solution (with < 1 ppm Triton X-100). Taken from Prakash et al. [119].

A thorough physical understanding of the mutual interactions between the continuous and dispersed phase is essential for the development of new technologies

with increased efficiency. Such understanding can be acquired by theoretical, experimental, and numerical methods. Due to the complexity of two-phase flows in general, the utility of theoretical methods is limited. Experimental methods can provide accurate point measurements by using hot-film and hot-wire anemometry, which have a very high time resolution. However, disadvantages are the low spatial resolution and the influence of the probe and probe support on the flow. Particle Image Velocimetry (PIV) can provide unsteady and multidimensional velocity fields by following the motion of small tracer particles (which are assumed not to influence the flow). However, application of PIV is limited to two-phase flows with low volume fractions (due to limited visibility at high volume fractions) and tracer particles will influence the occurrence of coalescence and breakup events of the dispersed phase. Numerical methods are a viable approach to study turbulent two-phase bubbly flows, thanks to the ongoing progress in the development of parallel computer systems and numerical algorithms. Especially Direct Numerical Simulation (DNS), where the fluid motion within and around each bubble is fully resolved, is effective in acquiring additional physical understanding. However, in most practical applications the ratio between the system size (large scale) and continuum scale (small scale) is many orders of magnitude. In such cases it becomes infeasible to fully resolve the fluid motion, and falling back to averaged flow equations becomes inevitable. Since these averaged flow equations by definition do not (fully) resolve the small-scale motions (e.g. viscous dissipation, individual bubble motion, interactions between bubbles, interaction between phases), these effects must be added to the averaged equations by so-called closure models. These closure models rely on the assumption that the small-scale motions are universal, and their quality has a strong influence on the applicability of the averaged flow equations. Although DNS is usually inefficient for practical applications, it can support the development of closure models by providing insight and data for the modeling of the small-scale effects on the averaged flow.

The last two decades much progress has been made in the application of DNS to dispersed two-phase flows consisting of multiple bubbles. Zhou and Pozrikidis [167, 168] applied the Boundary Integral Method (BIM) to study the flow of a periodic suspension of 1 to 12 two-dimensional droplets in a channel at a small Reynolds number (Stokes flow). A few years later, Loewenberg and Hinch [83, 84] applied BIM to study suspensions of up to 12 three-dimensional drops at a relatively high volume fraction (up to 30%) in a linear shear flow with viscosity ratios in the range of 0 to 5. However, in most applications the assumption of Stokes flow is not valid. Based on the front-tracking/finite-volume method of Unverdi and Tryggvason [157], Esmaeeli and Tryggvason [40, 41] started with the motion of rising bubbles (up to 324 two-dimensional or 8 three-dimensional bubbles) at $Re \sim 1 - 2$ in periodic domains. In [42] they extended this work up to 16 three-dimensional bubbles at $Re \sim 20 - 30$ for the lowest volume fraction. These simulations showed the mutual effects of bubbles on each other, and emphasize the importance of three-dimensional over two-dimensional simulations. Bunner and Tryggvason [14-18] studied the dynamics of up to 216 three-dimensional buoyant bubbles in periodic

domains to examine bubble rise velocity, induced velocity fluctuations, bubble distributions, and effects of bubble deformation. Thanks to methodological improvements and development of computational resources, Esmaeeli and Tryggvason [43] extended their earlier research to the buoyant rise of 48 bubbles at $Re \sim 100$ in a periodic domain. Lu et al. [88–92] were the first to perform DNS of turbulent bubbly downflow in a vertical channel with a finite-volume/front-tracking method. They considered cases of up to 72 bubbles (corresponding to a bubble volume fraction of 6%) with the undeformed bubble diameter equal to $d^+ \equiv \rho_c D u_\tau / \mu_c = 31.8$ (i.e., finite-size bubbles) at a friction Reynolds number $Re_\tau \equiv \rho_c H u_\tau / \mu_c = 127.3$ (based on the wall friction velocity u_τ and the half-channel height H). The Morton and Eötvös numbers were chosen such that the bubbles remained nearly spherical, and coalescence and breakup of bubbles were not allowed. The bubble/liquid mass density and dynamic viscosity ratios were taken equal to $\zeta = 0.1$ and $\lambda = 1$, respectively. The high shear force near the wall induces a lift force that drives the bubbles towards the core of the channel, thus leaving the wall regions bubble-free. This process keeps going until the core of the channel is in hydrostatic equilibrium and the bubble distribution is nearly uniform

Simultaneous with the front-tracking method the front-capturing method has been developed. Here the improvements of the volume-of-fluid (VOF) method by Lafaurie et al. [77], and the level-set (LS) method by Sussman et al. [144] are notable results. These methods were later combined by Sussman and Puckett [143] into the coupled LS and VOF (CLSVOF) method, which unites the strengths of both methods. Nevertheless, compared to the development and gained understanding obtained from the application of the front-tracking method to bubble-laden flows, front-capturing methods lag behind. However, front-capturing methods seem better suited when coalescence and breakup events are frequent [154]. Front-capturing methods can deal with such interfacial topology changes automatically, while front-tracking methods require regular remeshing of the interfacial marker particles. Bolotnov et al. [8] combined a stabilized Finite Element Method (FEM) with a LS approach to capture up to 32 air bubbles in a turbulent upflowing vertical water channel. Simulation results were averaged to obtain the liquid and gas mean velocity distributions, the local void fractions, the local turbulent kinetic energy, and the dissipation rate of the liquid phase. The influence of the bubbles on the turbulence was determined by comparing the two-phase with the corresponding single-phase turbulent channel flow results. By using a single LS function for all the bubbles, breakup and coalescence of air bubbles is (numerically) allowed.

The present DNS study employs the efficient multiple marker CLSVOF method of Coyajee and Boersma [27] and Kwakkel et al. [75] (see Chapter 2). Recently, this method has been extended with a coalescence/breakup module by Kwakkel et al. [76] (see Chapter 3), which is based on the film drainage model of Zhang and Law [166]. The aim of the present study is to determine the applicability of the CLSVOF method to perform DNS of dense turbulent dispersed two-phase flow, and in line with this the quantification of the influence of (coalescence/breakup of) bubbles on the turbulent flow. Based on previous mentioned studies, it is decided to simulate

a turbulent bubble-laden downflow in a vertical channel. A downflow ensures a bubble free wall region, which alleviates the absence of a bubble-wall interaction model. A dense flow ensures frequent collisions between bubbles, which allows the investigation of the influence of coalescence and breakup effects on the flow. To be able to quantify the influence of the (coalescence/breakup of) bubbles on the turbulent flow, a staged strategy is used. First, a benchmark simulation of a turbulent channel flow without bubbles will be performed. Next, bubbles will be added to the turbulent flow, initially without allowing any coalescence/breakup events. Finally, the coalescence/breakup model will be enabled, allowing the quantification of its individual influence on the turbulent flow. For the turbulent single-phase flow and the turbulent two-phase flow without coalescence/breakup experimental and/or numerical results are available, allowing validation of those numerical results. Unfortunately, due to initially unforeseen numerical issues during the simulation of the bubble-laden flow, the computational costs of these simulations turned out to be too high (see Section 4.5). Therefore, due to the limited amount of time available, no determinative statistics have been computed. Nevertheless, some preliminary results have been extracted from the data. Furthermore, no simulations with an enabled coalescence/breakup model have been performed.

The remainder of this chapter is organized as follows. First, in Section 4.2 details of the numerical method concerning the governing equations and simulation setup are given. Next, Section 4.3 and Section 4.4 present some preliminary results on turbulent single-phase and bubbly two-phase channel flows, respectively. Section 4.5 provides insight in the computational cost of the presented results. Finally, conclusions and recommendations are given in Section 4.6.

4.2 NUMERICAL METHOD

4.2.1 Governing equations

The motion of isothermal incompressible Newtonian fluids is given by

$$\nabla \cdot \mathbf{u} = 0, \quad (4.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \boldsymbol{\tau} + \mathbf{g}, \quad (4.2)$$

where $\mathbf{u} = (u, v, w)^T$ denotes the velocity vector, p the pressure, \mathbf{g} the gravity vector, and $\boldsymbol{\tau} = \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$ the viscous stress tensor. The density, ρ , and viscosity, μ , are assumed constant within each phase and defined by ρ_c , μ_c in the continuous and by ρ_d , μ_d in the dispersed phase. Since the fluids are considered immiscible, their mutual interface Γ is a material property of the flow and its motion can be described by Eq. (2.6)

$$\frac{\partial \Gamma}{\partial t} + \mathbf{u} \cdot \nabla \Gamma = 0, \quad (2.6)$$

where \mathbf{u} is the velocity of the fluid at the location of the interface. If the viscosity is regularized across the interface, the kinematic and dynamic interface conditions are given by Eqs. (2.7) and (2.8)

$$[\mathbf{u}]_{\Gamma} = 0, \quad (2.7)$$

$$[(-p\mathbf{I} + \boldsymbol{\tau}) \cdot \mathbf{n}]_{\Gamma} = -\sigma\kappa_{\Gamma}\mathbf{n}, \quad (2.8)$$

where $[\cdot]_{\Gamma}$ denotes a jump across the interface Γ (from the continuous phase to the dispersed phase), \mathbf{I} the unit tensor, \mathbf{n} the interface normal vector (which points from the continuous phase to the dispersed phase), σ the surface tension coefficient between the phases (assumed constant), and $\kappa_{\Gamma} \equiv -\nabla \cdot \mathbf{n}$ the interface curvature. Kang et al. [69] derived that when the viscosity is continuously varying across the interface, the derivatives of the velocity are also continuous, and (2.8) reduces to

$$[\nabla\mathbf{u}]_{\Gamma} = \mathbf{0} \quad (4.3)$$

$$[p]_{\Gamma} = \sigma\kappa_{\Gamma}. \quad (4.4)$$

A more detailed description of the mathematical model is given in Section 2.2.

4.2.2 Flow geometry and scaling

The governing equations are solved for a downflow in a vertical channel between two parallel walls. The streamwise, spanwise, and wall-normal directions are denoted by x , y , and z , respectively. The flow geometry and coordinate system are depicted in Fig. 4.2. The channel size, $L_x \times L_y \times L_z$, is defined as $6H \times 4H \times 2H$, where H is the half-width of the channel. At the walls ($z = 0$ and $z = 2H$) no-slip boundary conditions are used, and in the stream- and spanwise directions periodic boundary conditions are imposed. The latter can be justified when the computational domain is large enough to include the largest eddies in the flow, which is the case here.

The downward flow is driven by a streamwise pressure gradient, dp/dx , which can be derived from the macroscopic momentum balance. At a statistically steady state this balance is given by [88]

$$2\tau_w A - \frac{dp}{dx}V - \rho_{av}gV = 0. \quad (4.5)$$

Here τ_w is the average absolute wall shear stress, A the area of each wall, V the channel volume, and ρ_{av} the volume averaged density. Since $V = A \cdot 2H$, Eq. (4.5) can be simplified to

$$\tau_w = \left(\frac{dp}{dx} + \rho_{av}g \right) H. \quad (4.6)$$

As is common in fluid dynamics, Eqs. (4.1)-(4.2) are made dimensionless by a length scale \mathcal{L} , velocity scale \mathcal{V} , time scale $\mathcal{T} = \mathcal{L}/\mathcal{V}$, continuous phase density ρ_c , and continuous phase dynamic viscosity μ_c . Furthermore, to ensure that the fluid

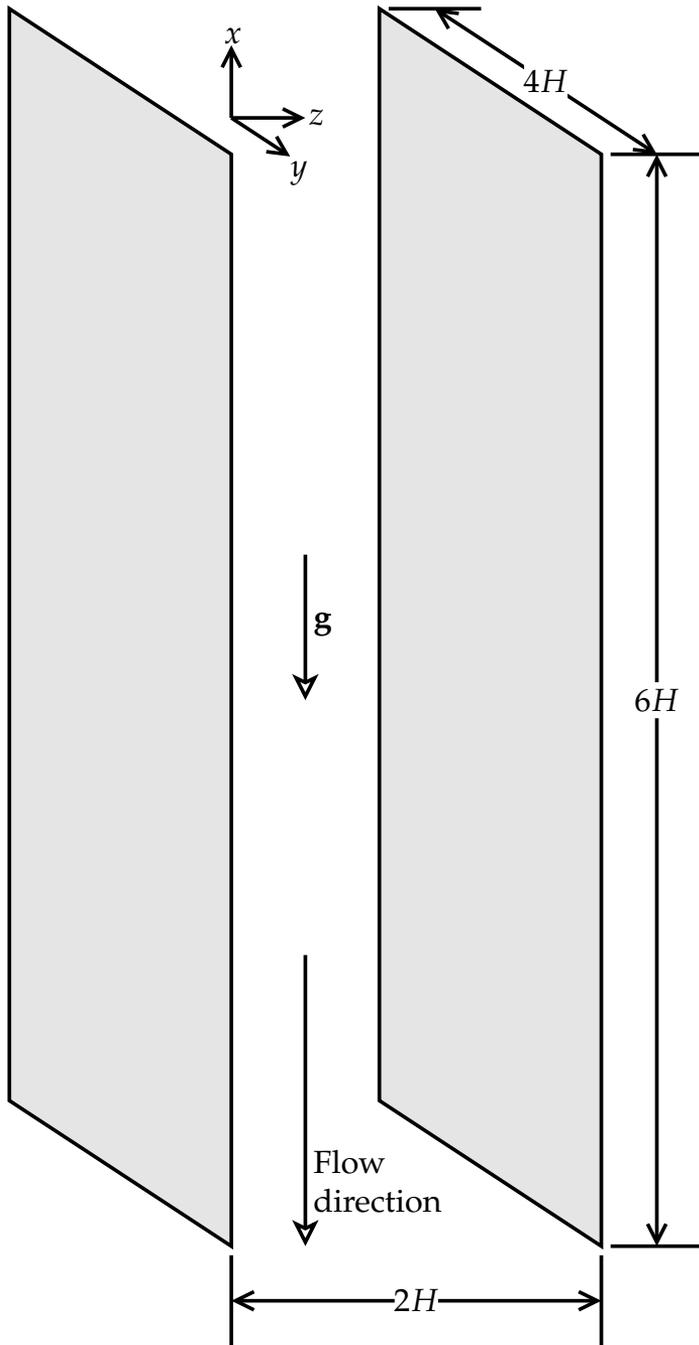


Figure 4.2: Flow geometry and coordinate system.

is at rest when the driving pressure gradient is zero, a modified pressure p_m with a correction for the hydrostatic contribution is used

$$p_m = p + \rho_{av} g x. \quad (4.7)$$

This leads to the following equations

$$\nabla \cdot \mathbf{u}' = 0, \quad (4.8)$$

$$\frac{\partial \mathbf{u}'}{\partial t'} + \mathbf{u}' \cdot \nabla \mathbf{u}' = -\frac{1}{\rho'} \nabla p'_m + \frac{1}{\rho' Re} \nabla \cdot \boldsymbol{\tau}' + \frac{1 - \rho'_{av}/\rho'}{Fr} \hat{\mathbf{g}}. \quad (4.9)$$

where $Re \equiv \rho_c \mathcal{U} \mathcal{L} / \mu_c$, $Fr \equiv \mathcal{U}^2 / g \mathcal{L}$, g denotes the absolute value of the gravitational acceleration ($\hat{\mathbf{g}}$ is a unit vector in the direction of the gravitational acceleration, so here $\hat{\mathbf{g}} = (-1, 0, 0)$), and $\boldsymbol{\tau}' = \mu' (\nabla \mathbf{u}' + (\nabla \mathbf{u}')^T)$ the viscous stress tensor. The primes indicate dimensionless variables, and are dropped for clarity for the remainder of this chapter.

Frequently used length scales to non-dimensionalize the equations are the half or full width of the channel, denoted by H and $2H$, respectively. Common velocity scales are the friction velocity, u_τ , and the bulk velocity, U_b . The friction velocity is defined by

$$u_\tau \equiv \sqrt{\frac{\tau_w}{\rho_c}}, \quad (4.10)$$

and characterizes the shear at the wall. The bulk velocity is defined by

$$U_b = \frac{1}{2H} \int_0^{2H} U(z) dz, \quad (4.11)$$

and characterizes the volume flow through the channel. Based on these scaling parameters, the following two Reynolds numbers can be defined

$$Re_\tau = \frac{u_\tau H}{\nu_c}, \quad (4.12)$$

$$Re_b = \frac{U_b 2H}{\nu_c}. \quad (4.13)$$

In this chapter the half-width of the channel and the friction velocity are used to scale the equations. Based on Eqs. (4.6), (4.7) and (4.10), by definition this corresponds with a dimensionless (modified) pressure gradient equal to $dp_m/dx = 1.0$.

The Reynolds number is set to $Re_\tau = 180$, which corresponds to a full-width bulk Reynolds number $Re_b \approx 5534$. The latter can be derived from the friction factor, C_f , for which Dean [31], based on experiments, proposed the following correlation

$$C_f = 0.073 Re_b^{-0.25}, \quad (4.14)$$

which can be complemented with the definition of C_f

$$C_f \equiv \frac{\tau_w}{\frac{1}{2} \rho_c U_b^2} = \frac{2u_\tau^2}{U_b^2} = \frac{8Re_\tau^2}{Re_b^2}. \quad (4.15)$$

Near the wall, where viscous effects are dominant, velocity and length are scaled by u_τ and $l^+ = \nu_c/u_\tau$, respectively. The resulting dimensionless variables are

$$u^+ \equiv \frac{\bar{u}}{u_\tau}, \quad (4.16)$$

$$z^+ \equiv z \frac{u_\tau}{\nu_c}. \quad (4.17)$$

In terms of (viscous) wall units, z^+ , the size of the channel is $1080 \times 720 \times 360$. This domain is larger than the so-called minimum turbulent channel of Jiménez and Moin [66], which is the smallest flow geometry which can adequately resemble a full-scale turbulent channel flow. The computational grid is taken as uniform, without any refinement in the wall region. To be able to resolve the viscous sublayer ($z^+ < 5$) with sufficient accuracy, the spatial resolution is chosen to be $H/h = 120$. With this configuration each cell measures 1.5 wall units, and the first grid point is located at 0.75 wall units from the wall. The viscous sublayer is therefore resolved by almost 4 grid points. Although this gives a high resolution in the core region and puts a limit on the computational time step, the uniform grid simplifies the CLSVOF interface advection. Furthermore, to be able to adequately capture a bubble at least 24 cells across its diameter ($D/h = 24$) is required [26]. For the considered domain the number of cells, $N_x \times N_y \times N_z$, is set to $720 \times 480 \times 240$, which makes the total number of cells equal to $N_t = 82.944 \cdot 10^6$.

The turbulent flow can be maintained by fixing the pressure gradient or the bulk velocity. The first approach is simple, since the pressure gradient is directly used in Eq. (4.9). As a consequence, the instantaneous bulk velocity will fluctuate a bit around the mean bulk velocity due to the finite size of the computational domain. The second approach, where the bulk velocity is kept constant, is only possible by adjusting the pressure gradient such that the momentum balance of Eq. (4.5) is exactly zero. Theoretically this is achieved if the pressure gradient is given the value of the integral of the shear force, which in dimensionless form reads

$$2 \left. \frac{\partial p_m}{\partial x} \right|_1 = \sum_{\text{walls}} \tau_w = 2 \left. \frac{\partial p_m}{\partial x} \right|_0 + 2 \left. \frac{\partial p_m}{\partial x} \right|_c. \quad (4.18)$$

Here the subscript 1 indicates the final corrected pressure gradient, the subscript 0 indicates the pressure gradient as used in the prediction step of the pressure-correction method (see Section 2.2.2) and the subscript c indicates the correction needed to ensure a fixed bulk velocity. However, if the velocity gradient for the computation of the total shear force is derived from an iteratively obtained velocity field, the momentum balance of Eq. (4.5) will not be zero up to machine precision. As a consequence, there will be slight errors in the derived pressure gradient and bulk velocity. This can be circumvented by computing the correction to the pressure gradient from

$$\left. \frac{\partial p_m}{\partial x} \right|_c = \frac{1}{\Delta t} \left(U_b - \langle u^{n+1} \rangle \right), \quad (4.19)$$

where the subscript c indicates the correction to the pressure gradient, U_b is the fixed non-dimensional mean bulk velocity, and $\langle u^{n+1} \rangle$ the streamwise velocity aver-

aged over the cross flow area at time level $n + 1$. For both approaches the correction to the pressure gradient is used to adjust the streamwise velocity via

$$u^{n+1} \Big|_1 = u^{n+1} \Big|_0 + \Delta t \frac{\partial p_m}{\partial x} \Big|_c, \quad (4.20)$$

where the subscript 0 and 1 indicate the velocity after the correction step of the pressure-correction method, and the corrected velocity with the prescribed bulk velocity, respectively. Note that Eq. (4.20) only adds a constant in space to the velocity field, which does not change the divergence of the velocity field and can be seen as an additional pressure-correction step [10].

4.3 TURBULENT SINGLE-PHASE CHANNEL FLOW

To be able to investigate the mutual interaction between the bubbles and the turbulent carrier fluid, as a reference a turbulent flow without bubbles has been simulated and analyzed. In this section, first the initialization of the turbulent flow is described. Next, the derived statistics are shown and compared with existing results. These are the results of Kim et al. [70], Moser et al. [101], and del Álamo and Jiménez [3], who simulated single phase turbulent channels for $Re_\tau = 180$, $Re_\tau = 178$, and $Re_\tau = 186$, respectively.

4.3.1 Flow initialization

The initial velocity field used for the single-phase turbulent flow was obtained from the simulation of a similar turbulent channel, which was computed with the same code on a coarser grid. The turbulence within this coarser simulation was initialized by a mean turbulent velocity profile with random velocity fluctuations (with a mean of zero) superimposed on it. This coarser simulation was run until turbulence was able to sustain itself and turbulence statistics reached an approximate steady state. Next, the coarse velocity field was interpolated on a finer mesh and run with a fixed bulk velocity ($U_b = 15.63$, as derived by Kim et al. [70]) for some time. The correlation of Dean [31], as introduced in Section 4.2.2, provides a slightly lower dimensionless bulk velocity

$$\frac{U_b}{u_\tau} = \frac{Re_b}{2Re_\tau} \approx 15.34. \quad (4.21)$$

This discord on the value of the bulk velocity for a flow with $Re_\tau = 180$, and the fact that theoretically $dp/dx = 1.0$ does lead to $Re_\tau = 180$, supports the decision to switch to a fixed pressure gradient at $t = 3.0$. This change in the application of the driving force for the turbulent channel is clearly visible in Figs. 4.3 and 4.4.

Due to the computational costs, see Section 4.5, the number of simulated time scales is kept relatively low. This decision does have some negative consequences for the averaged flow properties. To compute these averaged flow properties, 81 samples were taken during the last 8 non-dimensional time units. The start of this

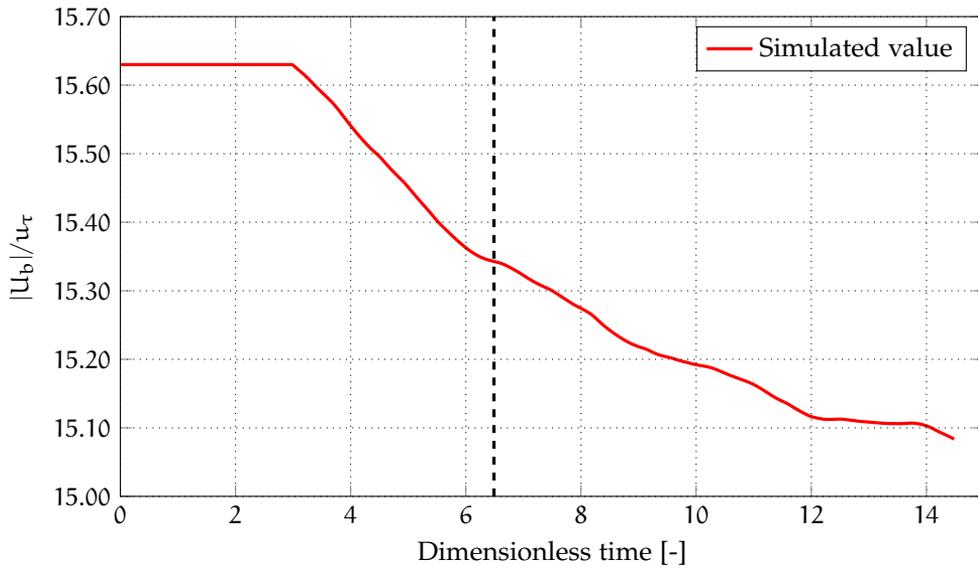


Figure 4.3: Development of the bulk velocity in time. The vertical dotted line indicates the start of the sampling period.

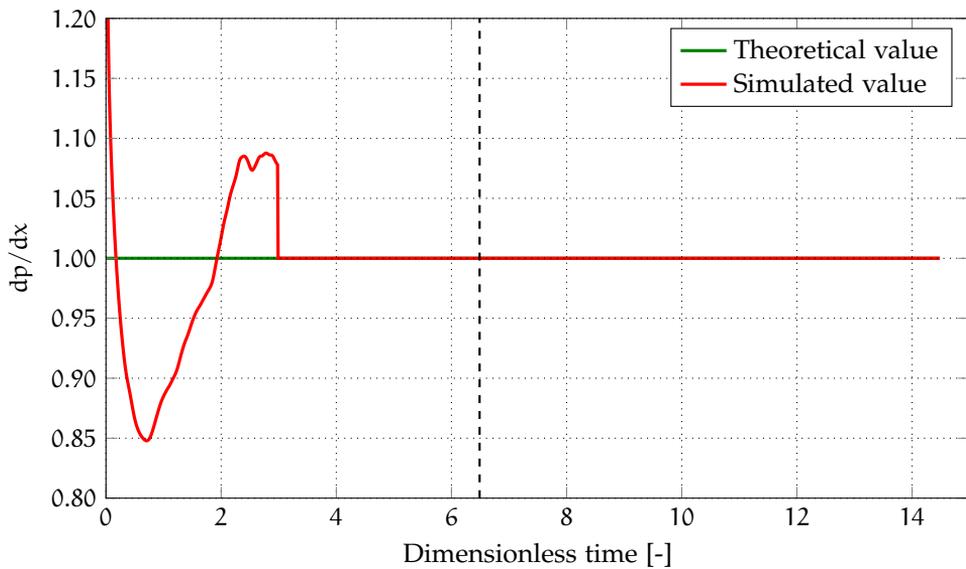


Figure 4.4: Development of the pressure gradient in time. The vertical dotted line indicates the start of the sampling period.

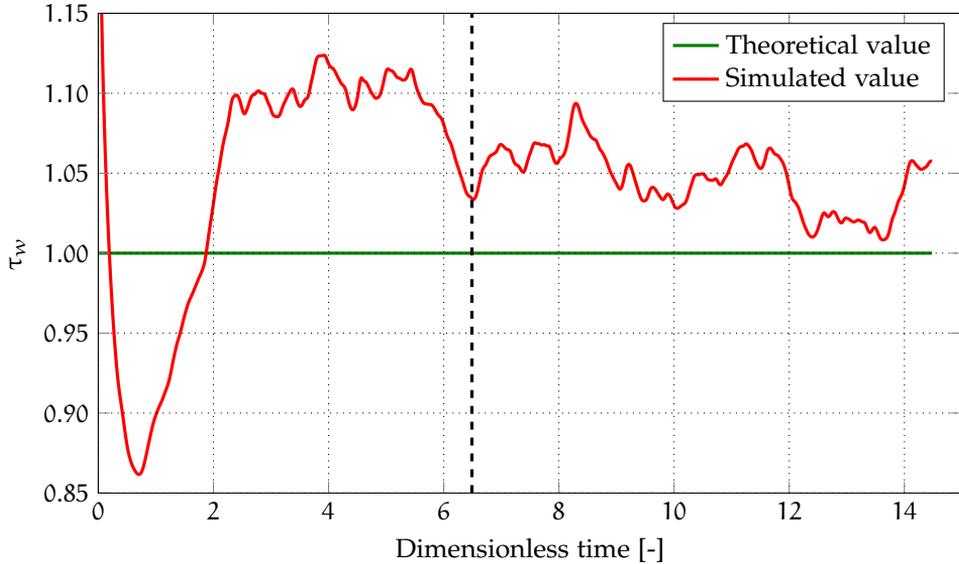
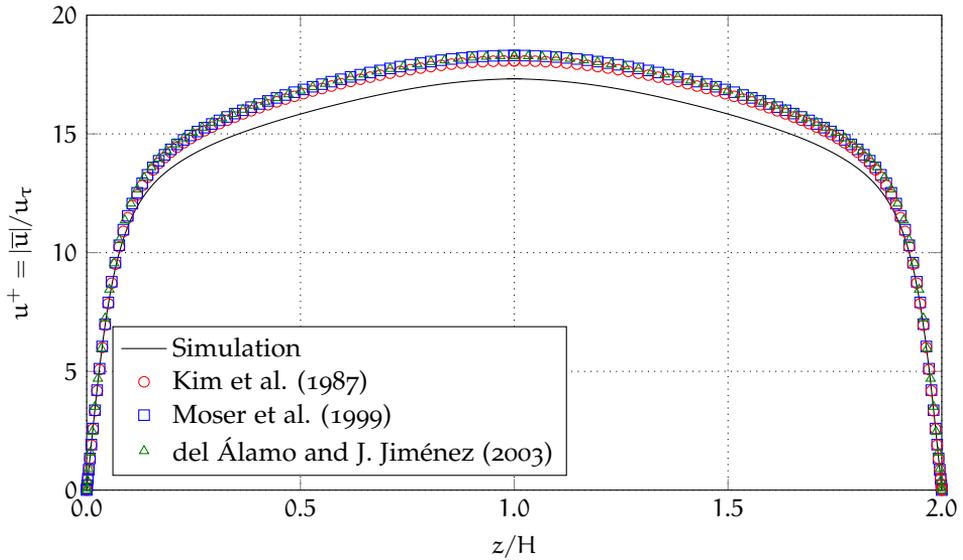


Figure 4.5: Development of the wall shear stress in time. The vertical dotted line indicates the start of the sampling period.

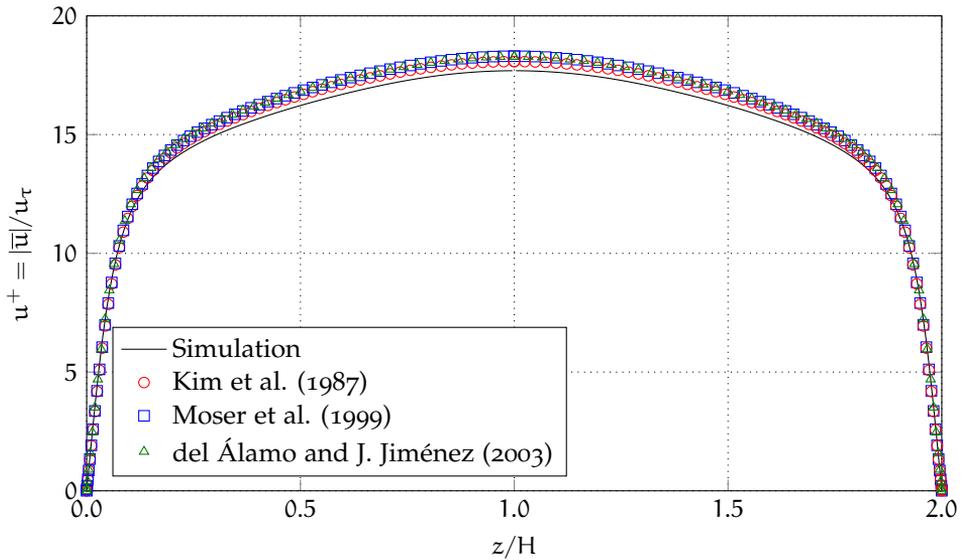
period is indicated by the vertical dotted line in the Figs. 4.3-4.5. Fig. 4.3 shows that the bulk velocity is still decreasing within the sampling time frame, and has not reached a fully steady state yet. Furthermore, Fig. 4.5 shows that the wall shear stress does also not fluctuate around the theoretical value of 1.0 yet. Although both profiles indicate that it is too early to start sampling, due to time constraints this is nevertheless done. To improve the quality of the statistics, averaging with respect to the center plane of the channel has been performed (based on mirror symmetry of the mean flow with respect to the centerline). Due to the poor convergence of the wall shear stress, the computed friction velocity ($u_\tau \approx 1.021$) deviates from its theoretical value ($u_\tau = 1.0$). Therefore, in the next sections all plots where u_τ is used to scale velocities, separate plots for the computed and theoretical friction velocities are presented.

4.3.2 Turbulence statistics

The mean velocity is computed by ensemble averaging the velocity field in planes parallel to the wall. Figs. 4.6a and 4.6b show the mean velocity profile, scaled by the computed and theoretical friction velocity, respectively. Both mean velocity profiles are substantially lower than the references, especially in the core region of Fig. 4.6a, which is scaled with the computed friction velocity.



(a) Mean velocity normalized by the computed friction velocity



(b) Mean velocity normalized by the theoretical friction velocity

Figure 4.6: Mean velocity profile, normalized by the friction velocity u_τ and shown as a function of the dimensionless channel width z/H .

Theoretically the normalized velocity profile near the wall is given by [128]

$$u^+ = \begin{cases} z^+ & \text{if } 0 < z^+ < 5, \\ \frac{1}{\kappa} \ln(z^+) + B & \text{if } 30 < z^+, \end{cases} \quad (4.22)$$

where $\kappa = 0.4$, and $B = 5.0$ (for a smooth wall). The region near the wall ($z^+ < 5$) is the viscous sublayer where viscous forces dominate, and the region further away from the wall ($z^+ > 30$) is the logarithmic layer where turbulent stresses dominate. The region between these two profiles ($5 < z^+ < 30$) is the so-called buffer layer. In this layer the viscous as well as the turbulent stresses are important. The profiles of Eq. (4.22) intercept each other at $z^+ = 11$, however they are only accurate in the ranges given above. Figs. 4.7a and 4.7b show a semi-logarithmic plot of the velocity profile near the wall, scaled by the computed and theoretical friction velocity, respectively.

Both results are compared with three references and the theoretical profiles of Eq. (4.22). The viscous sublayer matches the theoretical velocity profile, especially in Fig. 4.7a, where the computed friction velocity is used to normalize the velocity profile. In the logarithmic layer the theoretical profile matches best, while the three references are all somewhat higher.

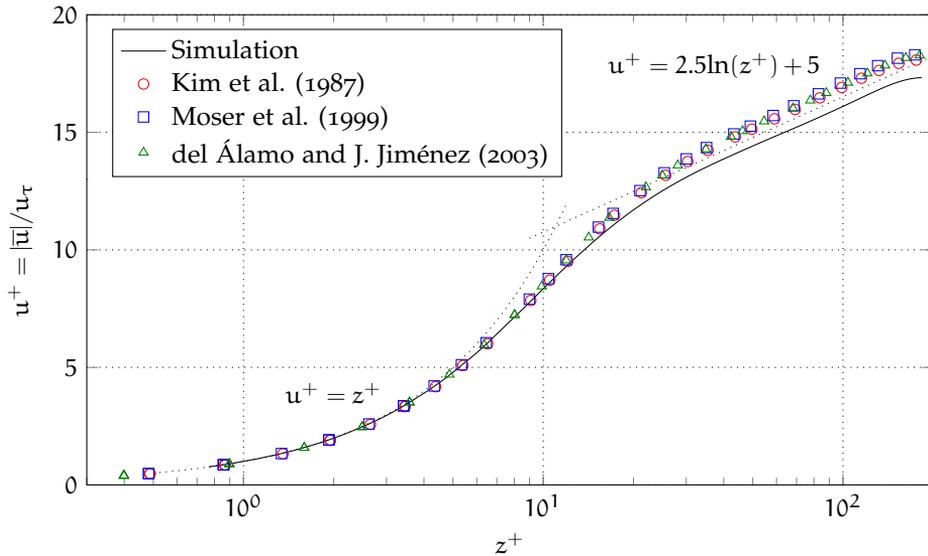
Figs. 4.8a and 4.8b show the profiles of the rms velocities, scaled by the computed and theoretical friction velocity, respectively. Both results are compared with three references for all three components of the rms velocity. Fig. 4.8a, which is scaled with the computed friction velocity, shows the best correspondence with the three references in the core region. This is in contrast with Fig. 4.8b, which is scaled with the theoretical friction velocity, where the best correspondence is obtained near the wall.

The total shear stress, τ_t , is the sum of the Reynolds and the viscous shear stress and given by

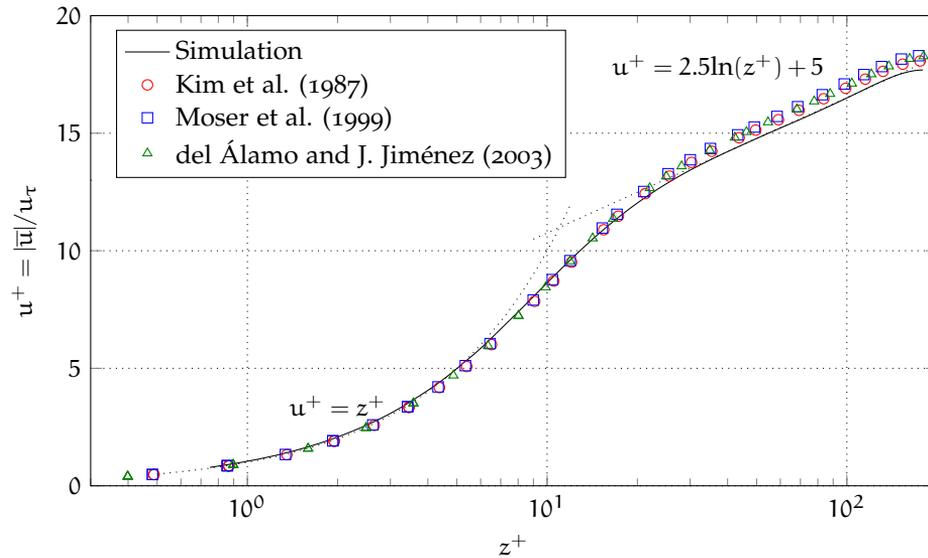
$$\tau_t \equiv -\overline{u'w'} + \nu_c \frac{\partial \bar{u}}{\partial z} = u_\tau^2 \left(\frac{z}{H} - 1 \right), \quad (4.23)$$

which shows that the total shear stress is a linear function of z . The profiles in Figs. 4.9a and 4.9b are normalized by the computed and theoretical u_τ^2 , respectively. Since the flow under consideration is a downflow, and by the definition of the orientation of the channel, the total shear stress is negative at the wall. As a reference, three results from literature and the theoretical linear total shear stress profile (dotted line) have been added. In Fig. 4.9a, where the computed friction velocity is used for scaling, the total stress at the wall is by definition equal to -1.0. When the theoretical friction velocity is used, as is done in Fig. 4.9b, the total shear stress in the entire channel is overestimated.

The current turbulence statistics of the channel provide enough confidence that the most essential physics is captured by the present numerical method and grid resolution. Of course the quality of the statistics will improve further when the simulation of the channel is continued over a much longer period. However, since the applied numerical solver is not optimized for single phase flows, combined with the limited amount of time available, it was decided to stop this single phase flow

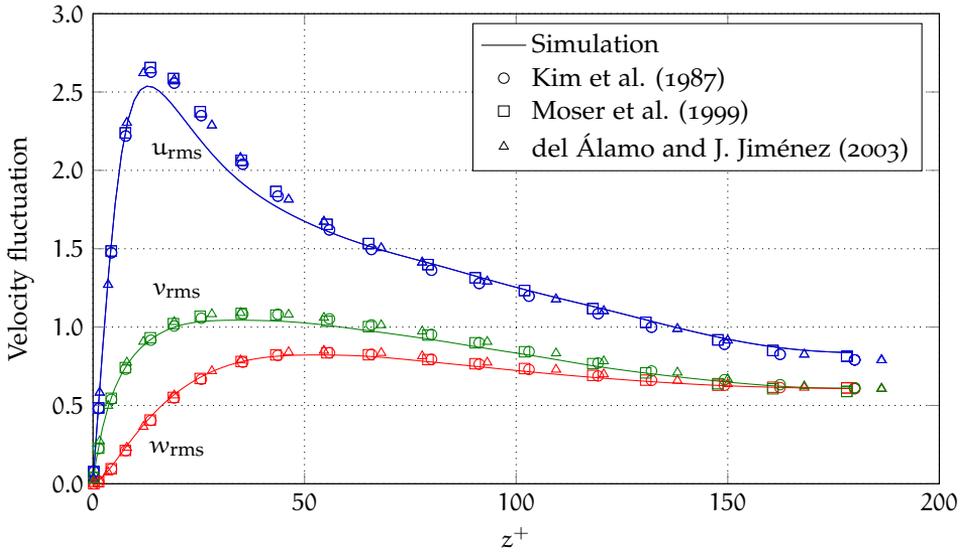


(a) Mean velocity normalized by the computed friction velocity

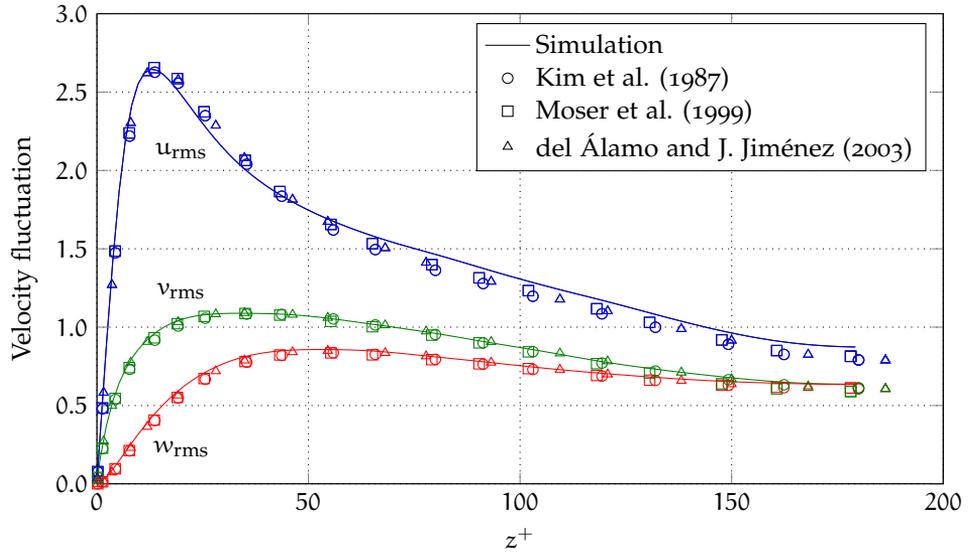


(b) Mean velocity normalized by the theoretical friction velocity

Figure 4.7: Mean velocity profile near the wall, normalized by the friction velocity u_τ and shown as a function of z^+ . The dotted curve corresponds to the theoretical velocity profile of the viscous sublayer, and the dotted line to the theoretical velocity profile of the logarithmic layer.

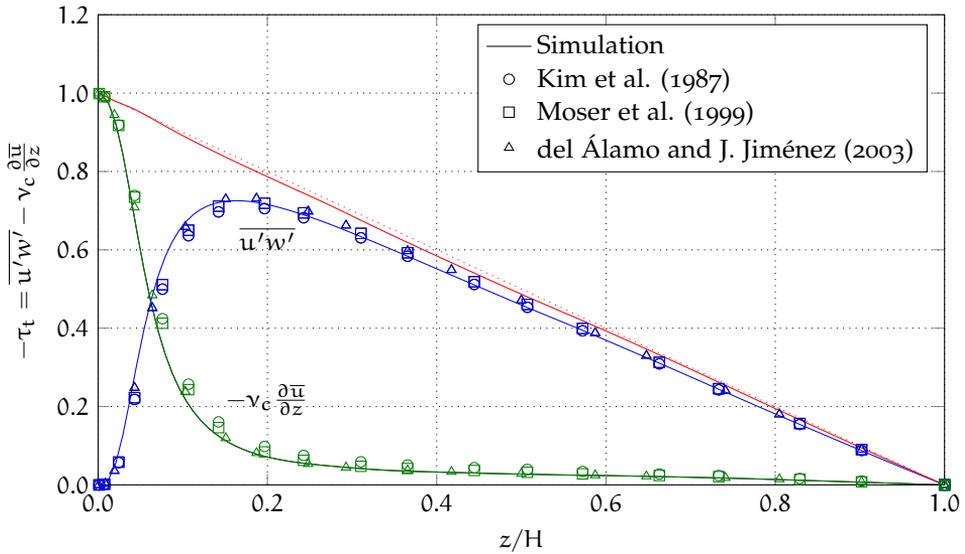


(a) Profile of the rms velocities normalized by the computed friction velocity

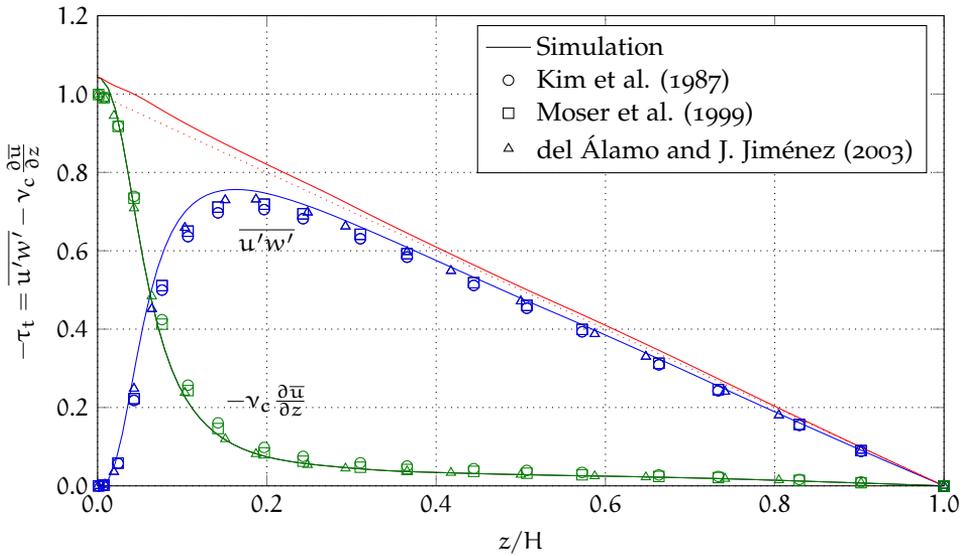


(b) Profile of the rms velocities normalized by the theoretical friction velocity

Figure 4.8: Profiles of the rms velocities normalized by the friction velocity u_τ . To validate the results, three references have been added.



(a) The computed friction velocity is used as u_τ



(b) The theoretical friction velocity is used as u_τ

Figure 4.9: Shear stress profiles normalized by u_τ^2 as a function of z/H . The symbols indicate the values from the references, and the dotted line the theoretical linear total stress profile.

simulation and add the bubbles. The characteristics and simulation results of this two-phase flow (without coalescence/breakup) are described in the next section.

4.4 TURBULENT BUBBLY TWO-PHASE CHANNEL FLOW

In this section the motion of 384 air bubbles in a turbulent downflowing vertical water channel is considered, which is comparable with the work of Lu and Tryggvason [88]. The most important differences with respect to that study are that here the entire system is bigger (384 bubbles compared to 72, and the total volume of the flow domain is about $2.6\times$ as big), and more realistic physical properties are considered (air bubbles in water, as already done by Bolotnov et al. [8]). Despite the fact that coalescence/breakup events can physically occur, and the ability of the current CLSVOF method to simulate them (see Chapter 3), for now these events are excluded a priori. Hence, the number of bubbles remains constant in time. Before more detailed simulations including these events are performed, the quality of the CLSVOF method to simulate many droplets in a turbulent channel should be quantified first. Up till now, the CLSVOF method has not been applied to bubble-laden flows as presented here, which make the simulations in this section of interest. The simulated liquid turbulence Reynolds number based on friction velocity and half channel width is $Re_\tau = 180$, which is state of the art.

Unfortunately, due to initially unforeseen numerical issues during the simulation of the bubble-laden flow, the computational costs of these simulations turned out to be too high (see Section 4.5). Therefore, due to the limited amount of time available, no determinative statistics have been computed. Nevertheless, some preliminary results have been extracted from the data. Furthermore, no simulations with an enabled coalescence/breakup model have been performed at all.

Section 4.4.1 specifies the used physical properties and model parameters of the simulated bubbly channel flow. Next, Section 4.4.2 explains how the bubbles have been added to the single phase turbulent channel. Finally, Section 4.4.3 discusses some preliminary results for the bubbly channel flow.

4.4.1 Bubbly flow specification

The bubbly flow system under consideration is a turbulent downflow of water with 384 dispersed air bubbles. The physical properties and model parameters for this bubble-laden two-phase channel flow are given in Table 4.1. As described in Section 1.3.2, the fluid equations are made dimensionless by characteristic length, velocity, and time scales. These scales are also given in Table 4.1, where the velocity scale (friction velocity, u_τ) is determined from $Re_\tau = 180$. Furthermore, the dimensionless parameters as used in the fluid equations are now given by

$$Fr = \frac{u^2}{g\mathcal{L}} = 2.64 \times 10^{-2}, \quad (4.24)$$

$$We = \frac{\rho_c u^2 \mathcal{L}}{\sigma} = 8.88 \times 10^{-2}. \quad (4.25)$$

The use of the droplet diameter and the droplet slip velocity (the velocity difference between the two phases) as characteristic length and velocity scales is physically more relevant. The slip velocity can be approximated by

$$u_s \sim \sqrt{\frac{\Delta\rho g D}{\rho_c}} \approx 9.90 \times 10^{-2} \text{ m s}^{-1}, \quad (4.26)$$

which gives $Fr_d \sim 1$ and $We_d \sim 0.1$, where the subscript d denotes the dispersed phase. Also the ratio of the droplet slip velocity and the turbulent velocity u_τ scale is of interest, which is approximated by $u_s/u_\tau \sim 3$. The channel size, $L_x \times L_y \times L_z$, is defined as $6H \times 4H \times 2H$, where H is the half-width of the channel. At the walls ($z = 0$ and $z = 2H$) no-slip boundary conditions, and in stream- and spanwise directions periodic boundary conditions are imposed. The latter can be justified when the computational domain is large enough to include the largest eddies in the flow, which is the case here. In terms of (viscous) wall units, z^+ , the size of the channel is $1080 \times 720 \times 360$. The computational grid is taken as uniform, and the number of cells, $N_x \times N_y \times N_z$, is set to $720 \times 480 \times 240$, which makes the total number of cells equal to $N_t = 82.944 \cdot 10^6$. With this configuration each cell measures 1.5 wall units, and the first grid point is located at 0.75 wall units from the wall. The viscous sublayer is therefore resolved by almost 4 grid points. Each bubble has a diameter of $D = 0.2H$, corresponding to 24 grid cells or 36 wall units over its diameter. The driving force for the downflow in the turbulent channel is a fixed pressure gradient, which is equal to $dp_m/dx = 1.0$.

Table 4.1: Physical properties and model parameters for the bubbly channel flow.

Air density	ρ_d	1.225	kg m^{-3}
Air viscosity	μ_d	1.8×10^{-5}	N s m^{-2}
Water density	ρ_c	1.0×10^3	kg m^{-3}
Water viscosity	μ_c	1.0×10^{-3}	N s m^{-2}
Air/water density ratio	$\zeta = \rho_d/\rho_c$	1.225×10^{-3}	
Air/water viscosity ratio	$\lambda = \mu_d/\mu_c$	1.8×10^{-2}	
Surface tension	σ	7.3×10^{-2}	N m^{-1}
Gravitational acceleration	g	9.81	m s^{-2}
Channel width	$2H$	1.0×10^{-2}	m
Droplet diameter	$D = 0.2H$	1.0×10^{-3}	m
Number of bubbles	N	384	
Volume fraction	α	3.35	%
Length scale \mathcal{L}	H	5.0×10^{-3}	m
Velocity scale \mathcal{U}	u_τ	3.60×10^{-2}	m s^{-1}
Time scale \mathcal{T}	H/u_τ	1.39×10^{-1}	s

4.4.2 Bubbly flow initialization

At the time the single-phase flow, as described in Section 4.3, approached (nearly) statistical convergence, 384 bubbles were added to the channel. The bubbles were placed in a regular $12 \times 8 \times 4$ array configuration, where the precise bubble locations were slightly perturbed (up to $\pm 0.015H$ with respect to the array position) to enhance bubble interactions. Each bubble has its own local marker functions (see Section 2.3.1), which prevents bubble merging by numerical (artificial) coalescence [26]. Initially the bubbles are spherical, which allows an accurate initialization of the LS and VOF functions. The turbulent bubbly flow simulation is restarted from the single-phase velocity field, without any special adjustments. Although the properties (density and viscosity) at the location of the bubbles are changed instantly (no ramping down, as was done by Bolotnov et al. [8]), no instabilities were observed. Initially the bubbles will have the same velocity as the continuous phase, but over time the velocities of the continuous and dispersed phases will deviate.

4.4.3 Preliminary results

To obtain an understanding of the dispersed two-phase flow, a snapshot of the bubble distribution and the streamwise velocity on a cross-section (placed at $y = 2.75$ and scaled with the friction velocity) of the flow are shown in Fig. 4.10. Although its hard to fully understand the distribution of the bubbles from this figure, the boundary layers on both walls are clearly visible. Fig. 4.11 depicts a close-up of the streamwise flow around a few bubbles cut by the slice. The black lines indicate the position of the bubble interfaces. All fluid moves downwards, the dispersed bubbles however at a lower velocity than the surrounding continuous phase (due to buoyancy effects). When a bubble passes, the continuous phase liquid has a lower effective area to flow through. This causes that the streamwise velocity halfway around a bubble is slightly higher than the average continuous phase velocity. Due to the fact that water has a higher viscosity than air, this increased lateral velocity also speeds up the air within the bubble. Furthermore, since the bubbles rise relatively to the continuous phase, a wake is created behind the bubble.

Unfortunately only less than one dimensionless time scale (H/u_τ) has been simulated (see Section 4.5), which makes it impossible to derive any determinative conclusions. Nevertheless, Fig. 4.12 shows the volume fraction distribution in time, where the last time level is indicated by the blue line. The red dashed line is a theoretical volume fraction distribution based on hydrostatic equilibrium. Due to the limited simulation period and expected symmetry, averaging with respect to the wall-parallel centerplane has been performed (assuming flow symmetry). For (nearly) undeformed bubbles, a downflow stimulates a bubble migration to the core of the channel (and a bubble free wall region), until hydrostatic equilibrium and a nearly uniform bubble distribution is obtained. The volume fraction in the core region, α_c , at hydrostatic equilibrium can be determined from

$$\alpha_c = \alpha + \frac{dp_m/dx}{g\Delta\rho} = 3.35 + \frac{1.0}{9.81 \cdot 998.775} \frac{\rho_c u_\tau}{H} = 3.35 + 2.65 = 6.0\%. \quad (4.27)$$

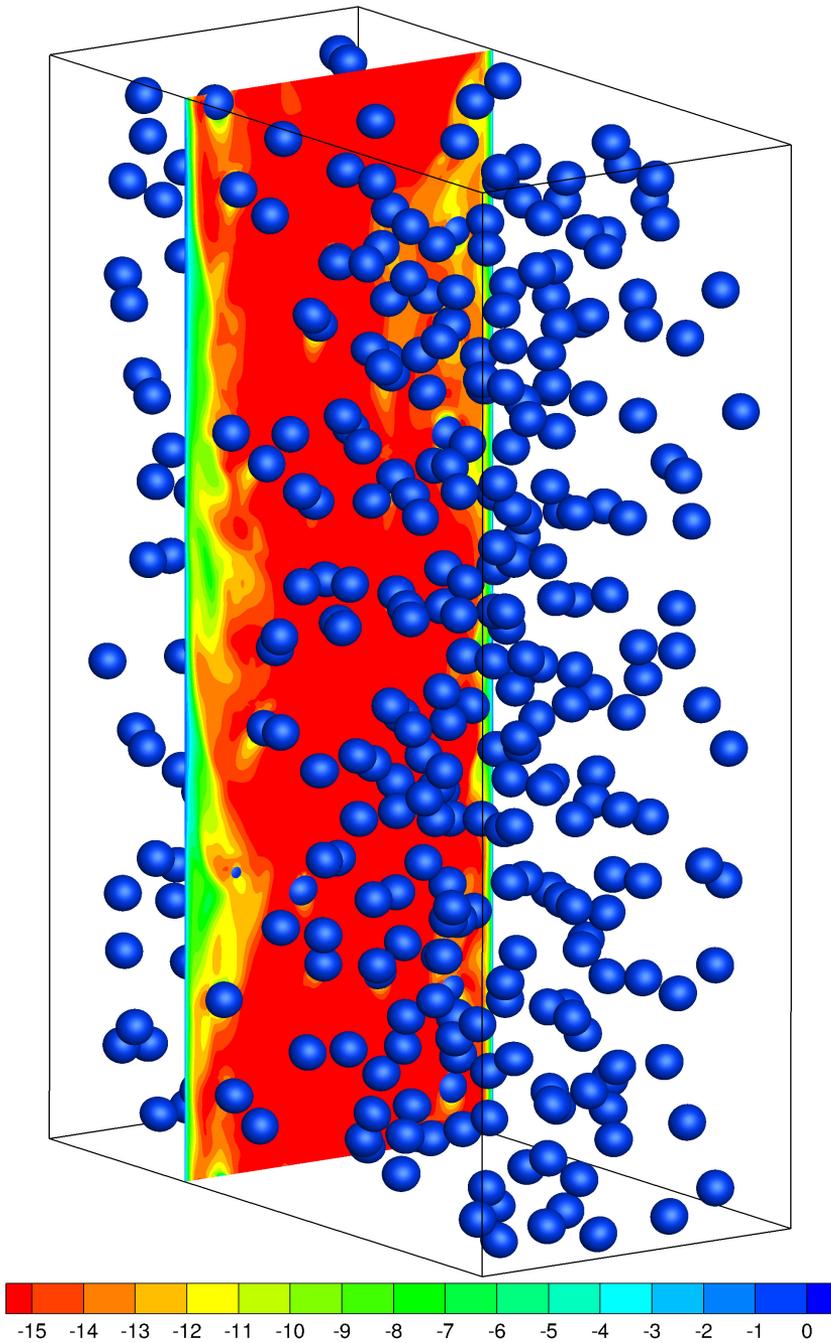


Figure 4.10: Bubble distribution in the turbulent channel. A slice is placed at $y = 2.75$, which shows the streamwise velocity. The velocities are scaled with the friction velocity, and values are as given in the legend.

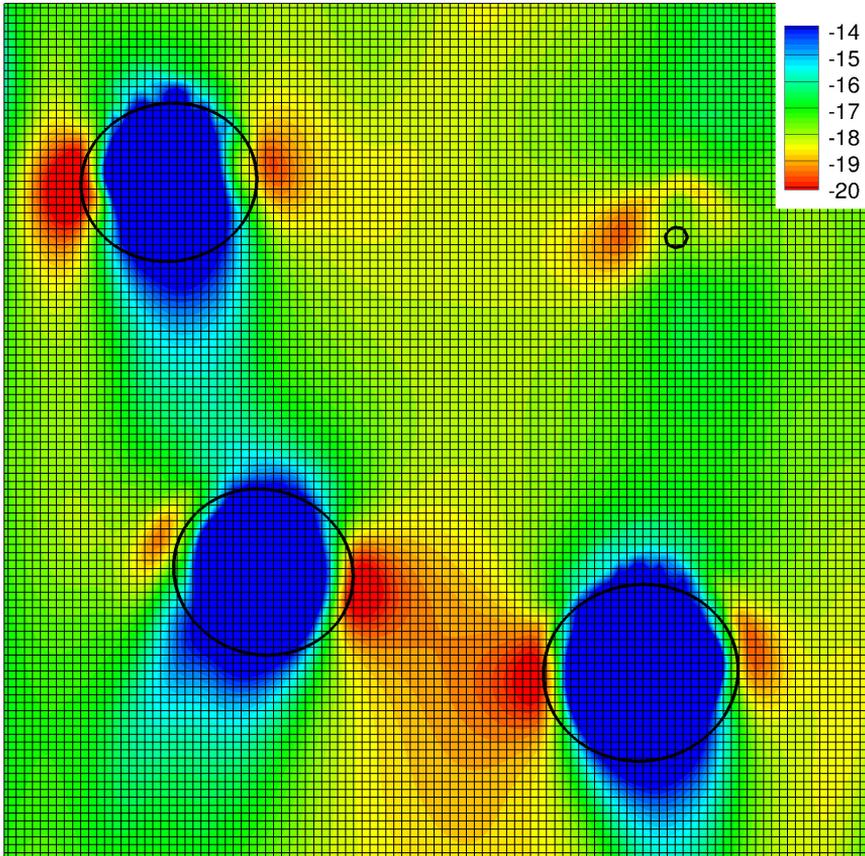


Figure 4.11: Closeup of the slice at $y = 2.75$, where the coloring indicates the stream-wise velocity as given by the legend. Sliced bubbles are indicated by the thick black contours, and the computational grid is shown to the size of the smallest resolved scales.

The thickness of the bubble free region near the wall, denoted by δ , can now be determined from

$$\delta/H = 1 - \frac{\alpha}{\alpha_c} = 0.4411. \quad (4.28)$$

Initially four peaks with an increased volume fraction are visible, which collapse over time towards the blue line (the last simulated time level). Although the dashed red line is only a theoretical limit, it is expected that at steady state the simulated volume fraction at the core will reach its theoretical value of 6.0%.

Fig. 4.13 shows the averaged velocity profiles across the channel for both phases at the last simulated time level. As a reference the single-phase velocity profile is given. To obtain these velocity profiles averaging over wall parallel planes has been performed. Since the volume fraction of the dispersed phase is not that high, the dispersed phase profile has been binned in ten equal bins.

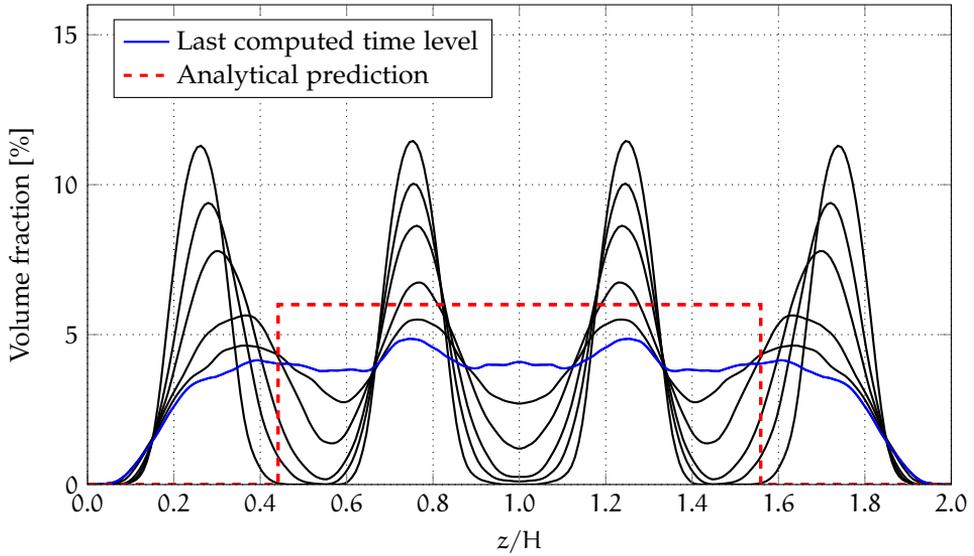


Figure 4.12: Volume fraction distribution over time, where the last simulated time level is indicated by the blue line. The dashed red line indicates a theoretical distribution, which is based on hydrostatic equilibrium.

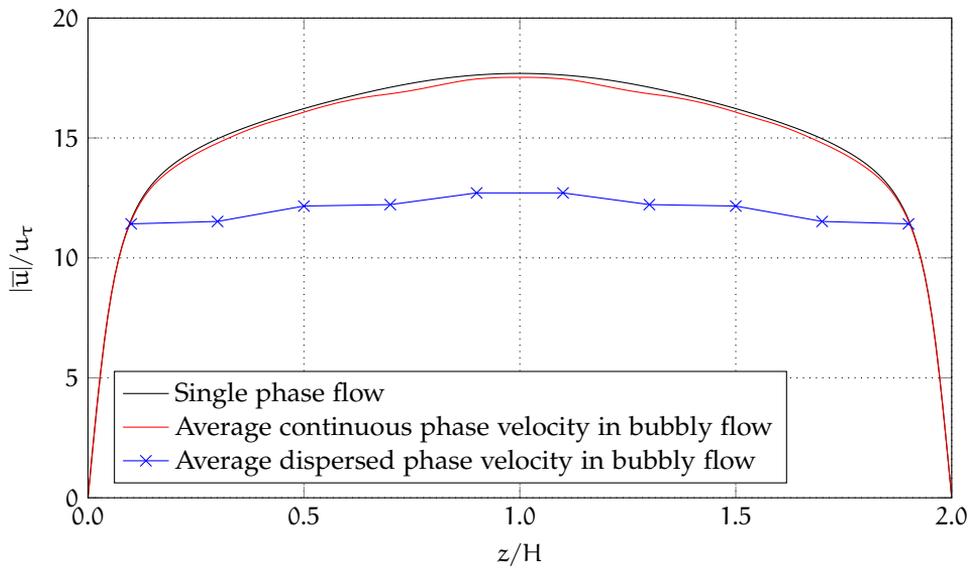
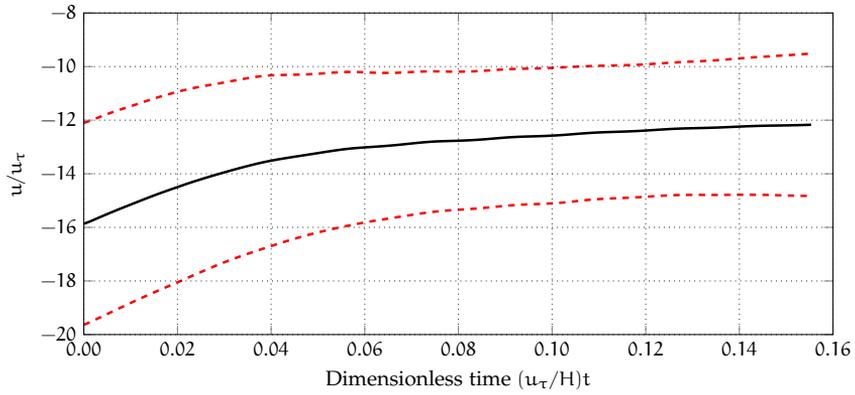
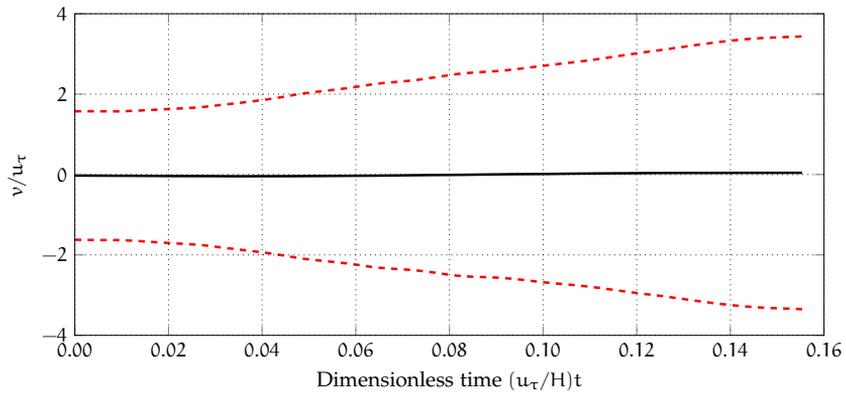


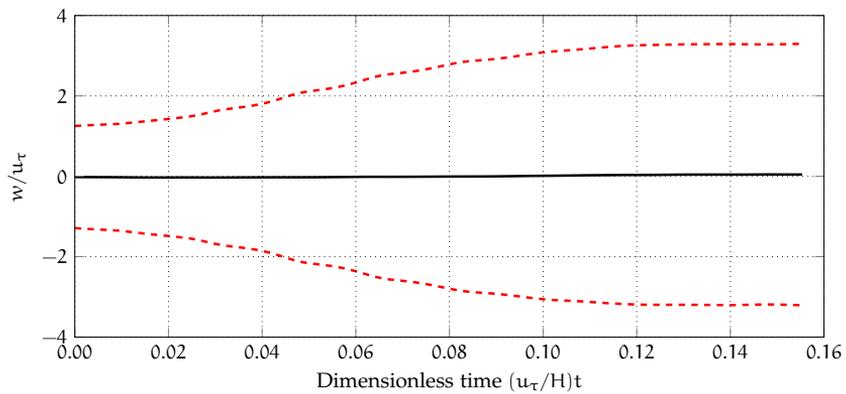
Figure 4.13: Averaged velocity profiles across the turbulent channel for both phases at the last simulated time level, compared to the single phase velocity profile.



(a) Streamwise velocity



(b) Spanwise velocity



(c) Wall-normal velocity

Figure 4.14: Evolution of the mean velocity of the bubble centroid (black lines), the dashed red lines indicate twice the standard deviation (95% confidence interval) among the bubbles.

As expected [88], the average bubble velocity is nearly uniform across the channel. Based on the relative velocity between the dispersed and continuous phases in the core of the channel, the bubble Reynolds number can be approximated as

$$Re_d = \frac{\rho_c D |u_c - u_d|}{\mu_c} \approx 0.96 Re_\tau = 172, \quad (4.29)$$

where $|u_c - u_d|$ is the relative velocity between the continuous and dispersed phase.

Fig. 4.14 shows the evolution of the mean velocity of the bubble centroid (black lines), the dashed red lines indicate twice the standard deviation (95% confidence interval) among the bubbles. The streamwise mean bubble velocity (and the width of the 95% confidence interval) has not converged yet. The spanwise and wall-normal mean bubble velocities fluctuate (as expected) around zero. The 95% confidence interval of the wall-normal bubble velocity seems more converged than the spanwise, probably due to the wall interaction.

4.5 COMPUTATIONAL COSTS

As mentioned in the previous sections, the computational costs of both single-phase and two-phase bubbly turbulent flow simulations are limiting the applicability of the current method. However, if the efficiency of the method can be improved, a very promising method is obtained which can provide insight in many flows of physical interest. Especially the addition of a coalescence and breakup module within an interface resolving method are a novelty. Therefore, by identifying and removing the largest bottlenecks of the current method, an undeveloped area of research becomes within reach. In this section the main issues are characterized, but also supplemented with possible remedies. First, the specifications of the used computational resources are given. The computational cost to simulate a flow for a certain amount of dimensionless time depends on the time step size, the wall clock time per time step, and the number of available processing elements (PEs, which are also denoted as cores). Since the number of PEs was fixed for both simulations, this choice is discussed in the first section on computational resources. Next, the time step size and the breakdown of the computation time per time step are discussed in more detail in separate sections.

4.5.1 Computational resources

The simulations were performed on Lindgren, which is a Cray XE6 system that is located at KTH in Stockholm, Sweden. The system is based on AMD Opteron 12-core Magny-Cours (2.1 GHz) processors and Cray Gemini interconnect technology. It consists of 1516 compute nodes and 10 service nodes, where the latter are used as login environment and internal services. Each node has 24 cores divided between 2 sockets, and is equipped with 32 GB DDR3 memory.

To balance the load as equally as possible over the available cores, the channel is parallelized in three dimensions where each subdomain is a cube. Therefore, it was decided to use 16 nodes, which provide 384 cores and a total memory of 512

GB. The number of cores is subdivided as $12 \times 8 \times 4$, which gives that each cube corresponds to $60 \times 60 \times 60 = 2.16 \cdot 10^5$ grid cells. Since each variable array has one halo cell on each side of the cube (used to synchronize with neighbors), and a double precision variable requires 8 bytes, the total amount of memory for each double precision variable is only 1.82 MB per core. Although the total amount of memory per node (32 GB) is shared by 24 cores, the required amount of memory per double precision variable (0.043 GB) is by far sufficient. The same parallelization was used for both single-phase and two-phase simulations, which makes mutual performance comparisons easier and more reliable.

Due to the master/slave technique, used for the parallelization of the bubble related routines, optimally the number of bubbles should be a multiple of the number of cores. Less bubbles would be equally expensive in terms of wall clock time (at least for interface advection and interface reconstruction), since these steps are done in parallel. To minimize the computational costs, and since this number of bubbles is already innovative, the total number of bubbles is set to 384.

4.5.2 Time step size

The wall clock time spend to reach a certain dimensionless time level strongly depends on the time step size (which determines the required number of time steps). Within the CLSVOF method the timestep is dynamically determined as described in Section 2.2.2. For single-phase flow only the convective and force time step constraints are important. Therefore, instead of a dynamic determination of the time step size, for simplicity a fixed time step of $\Delta t = 1 \cdot 10^{-4}$ was used. For single-phase flow this time step size ensured stability and accuracy. It is expected that in two-phase flows the surface tension time step constraint, which is given by Eq. (2.12)

$$\Delta t_s = \frac{1}{2} \sqrt{\frac{We \max(\zeta, 1)}{8\pi}} h^{3/2}, \quad (2.12)$$

is due to its $h^{3/2}$ scaling usually the most restrictive for fine grids. However, analysis of the dynamically determined time step constraints, see Fig. 4.15, shows that the convective time step constraint turned out to be by far more strict during the two-phase simulation. This unexpected behavior is caused by large velocity fluctuations, which are depicted in Fig. 4.16. The flow field remains divergence-free, but due to the high velocities the time step becomes occasionally very small to satisfy the CFL time step criterion for numerical stability. After analyzing the momentum and velocity fields in more detail, it was discovered that these large velocity fluctuations appear only within (some) bubbles. As described by Desjardins and Moureau [33], discretization inconsistencies in the discretization schemes for advective mass and momentum transport can give rise to spurious errors in the velocity field, especially in the presence of strong shear at interfaces and high density ratios. Fig. 4.17 shows the streamwise momentum (left) and velocity (right) fields for a wall-normal slice through the channel. The black contours indicate the interface locations of sliced bubbles. Any artificial momentum transfer from the continuous to the dispersed phase, where the density is much lower, would res-

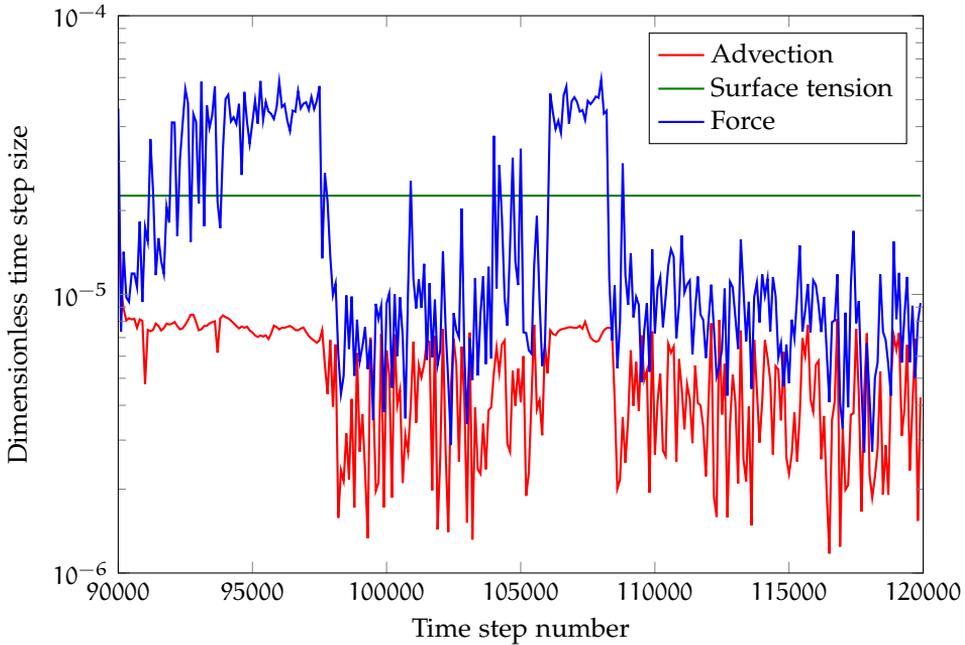


Figure 4.15: Time step restriction during the two-phase flow simulation as a function of the time step number.

ult in extreme velocities inside the dispersed phase, as can be seen in this figure. These observations strengthen the idea that the numerical instabilities are caused by an inconsistency in the discretization of the advection terms in the equations for the LS and VOF functions (mass) on the one hand, and the momentum equations on the other hand. While the advective terms in the equations for the LS and VOF functions are discretized in conservative form, the advective term in the momentum equation is discretized in non-conservative form. Consequently, the total momentum flux over the boundaries of a computational cell based on the mass flowing in/out of this cell can slightly differ from the momentum flux as used in the momentum equations. Two approaches that have been proposed in recent literature are (1) the assurance of a discretely consistent transport of mass and momentum by matching momentum fluxes [33, 49, 123, 125, 158], and (2) the use of velocity extension from the higher to the lower density fluid [33, 147]. From these two the first is the most elegant solution, and its implementation in the current CLSVOF method should not be difficult.

4.5.3 Time step breakdown

Fig. 4.18 shows the averaged time step breakdown for single-phase and two-phase flows. For both simulations around 85% of the wall clock time is spend in the pres-

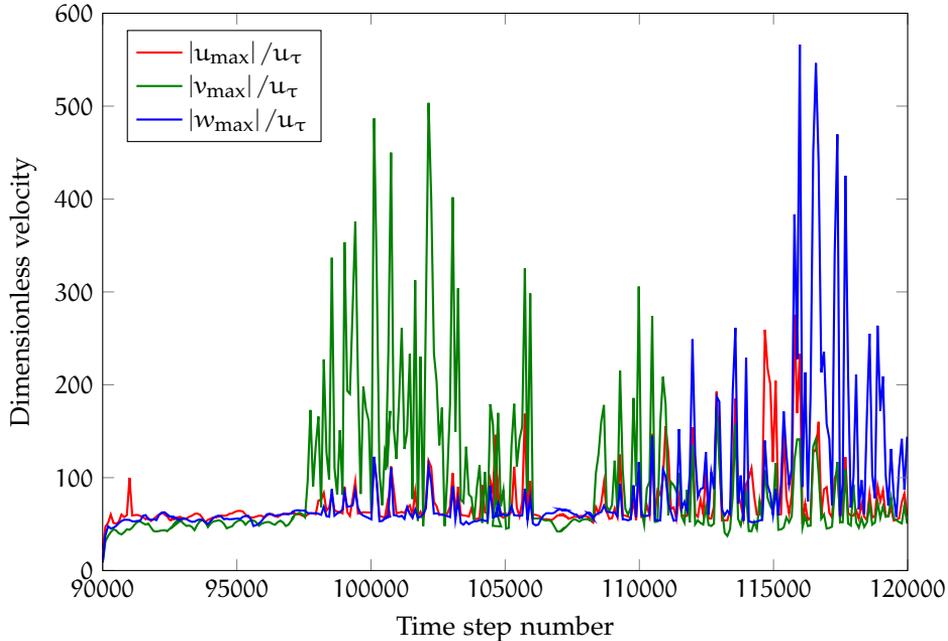


Figure 4.16: Absolute maximum velocities during the two-phase flow simulation as a function of the time step number.

sure solver. However, in absolute terms the pressure solver is much more expensive for the two-phase flow (+35.6s). This huge increase in workload is likely related to the minor instabilities caused by momentum transfer across the interfaces. The iterative solvers for the velocity are almost unaffected by the addition of the droplets (+0.2s). The advection time of the 384 bubbles is discounted in 'other', but its computational cost is limited (+3.1s). This is a direct result of the master/slave technique, which has been designed to improve the load-balancing for bubble-laden systems. Therefore, to improve the efficiency of the two-phase flow simulations, the largest gain will be achieved by making the pressure solver more efficient.

Fig. 4.18 shows that most time is spent in the pressure solver, so a more detailed investigation of this computational step is justified. Fig. 4.19a shows the number of iterations and wall clock time of the pressure solver with respect to the time step number during the two-phase flow simulation. What stands out is that the number of iterations and thereby also the wall clock time per time step of the pressure solver steadily increases. The deflation method, which uses (fixed) subdomain deflation vectors, should decrease the required number of iterations. However, the increasing number of iterations indicates that the deflation method becomes less effective over time. At the start of the simulation the bubbles are located in an almost regular array, and over time the bubbles distribute more randomly over the channel (see Fig. 4.12). Possibly this development, or that an increasing number of bubbles is influenced by the numerical instabilities related to the artificial momentum transfer

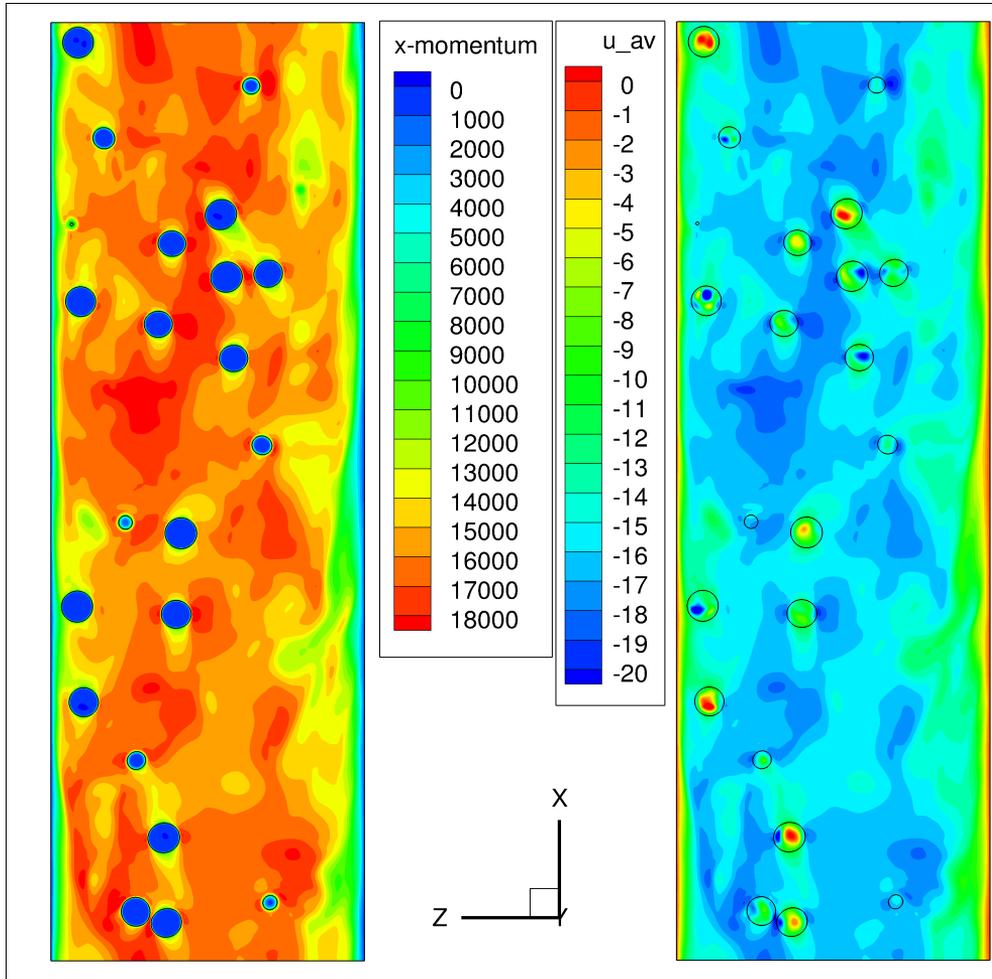


Figure 4.17: Momentum (left) and velocity (right) fields on a wall-normal slice through the channel. The black contours indicate the interface locations of sliced bubbles.

across interfaces, cause the deflation method to become less effective throughout the two-phase simulation. As a comparison, Fig. 4.19b shows the number of iterations and wall clock time of the pressure solver with respect to the time step number during the single-phase flow simulation. This plot underlines the large increase of computational cost by the addition of bubbles to the flow.

Fig. 4.20 shows the number of iterations and wall clock time of the velocity solvers with respect to the time step number during the two-phase flow simulation. These figures indicate that the wall clock time spend by the velocity solvers is rather insensitive to the large velocity fluctuations shown in Fig. 4.16. However, Fig. 4.20b

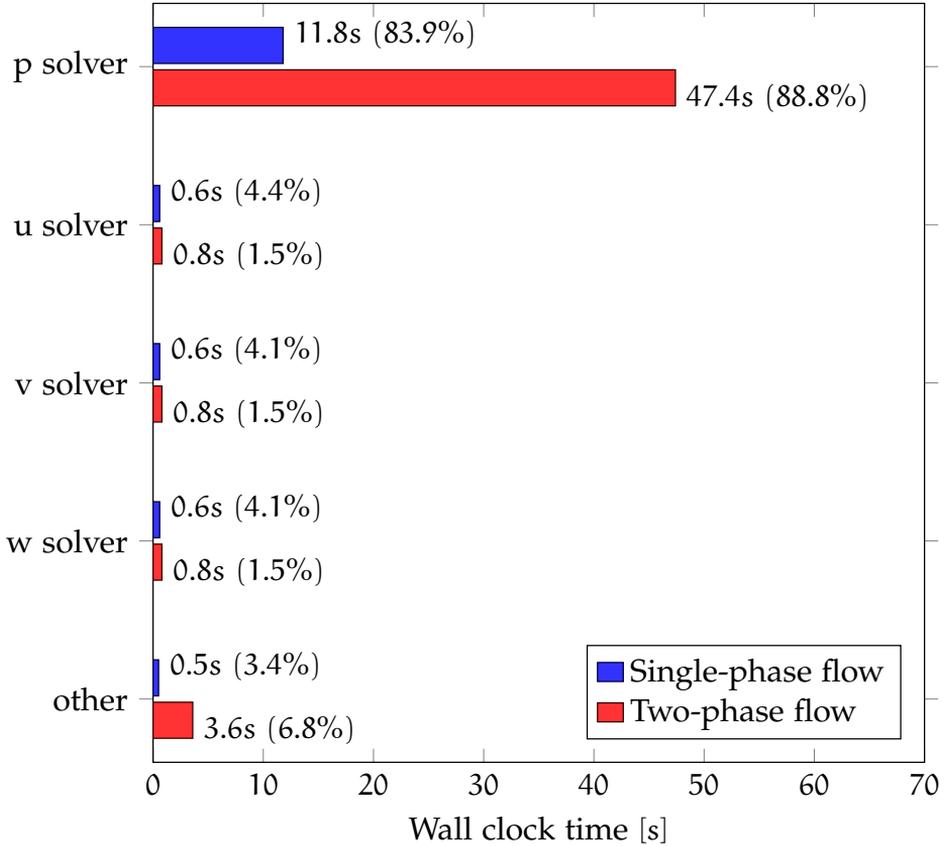
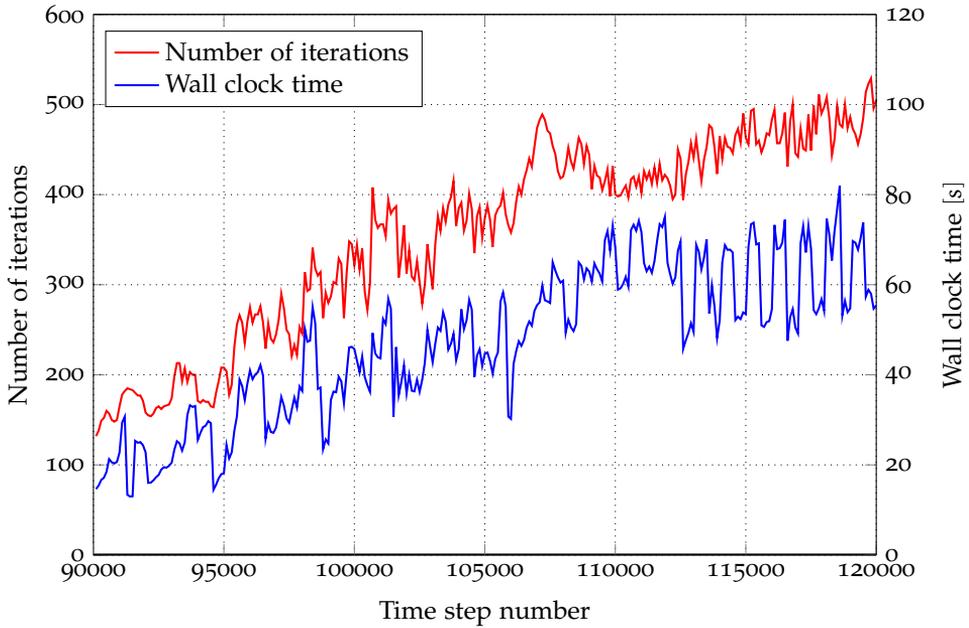


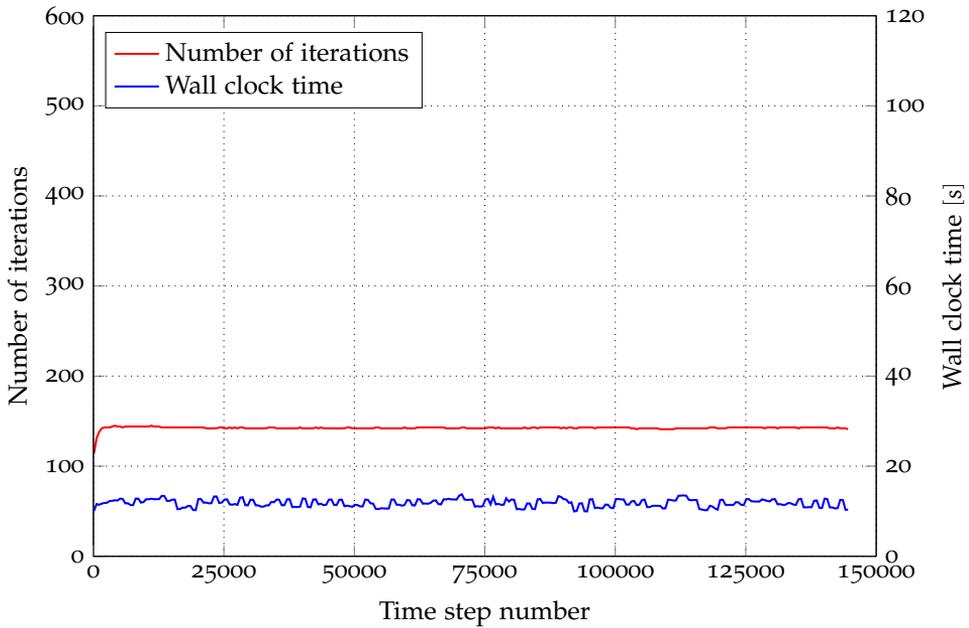
Figure 4.18: Averaged time step breakdown for single-phase and two-phase flow simulations. The wall clock time spend on interface advection of the bubbles is incorporated in 'other'.

shows a small increase of the number of iterations around time step 98000, probably caused by the corresponding large spanwise velocity fluctuations.

As can be observed in Figs. 4.19a and 4.20, strong stepwise fluctuations in the wall clock time occur for the last 8000 time steps. To check if these fluctuations are caused by the computational system, and therefore not by our own code, scaling with respect to the wall clock time per iteration of the pressure solver is done. This scaling is based on the assumption that the number of floating point operations during one iteration is constant, and therefore the wall clock time per iteration should be constant as well. This scaled plot of the wall clock time is shown in Fig. 4.21. Since the stepwise fluctuations have disappeared, the assumption that these are caused by the computational system is correct.

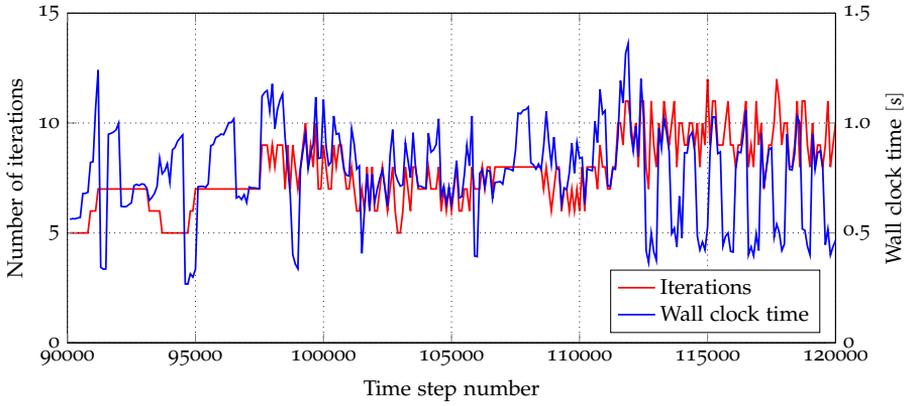


(a) Two-phase pressure solver.

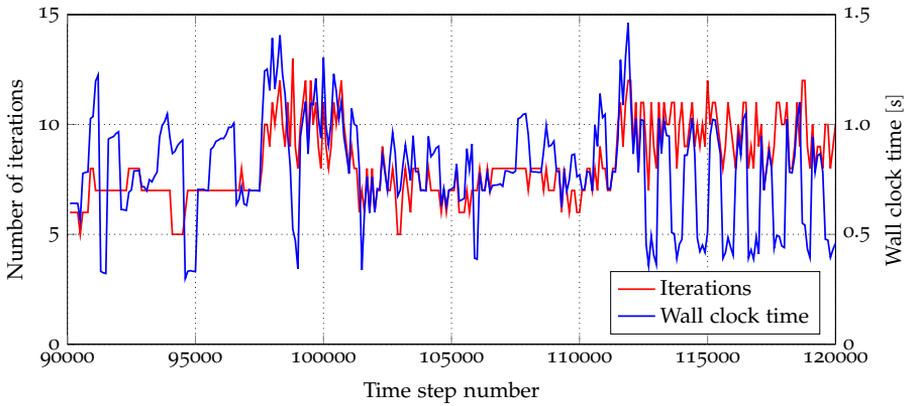


(b) Single-phase pressure solver.

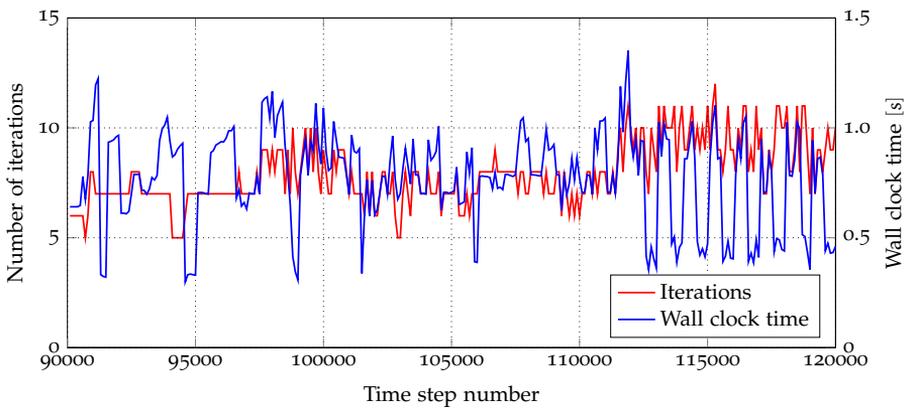
Figure 4.19: Number of iterations and wall clock time of the iterative pressure solver with respect to the time step number.



(a) *u*-velocity solver.



(b) *v*-velocity solver.



(c) *w*-velocity solver.

Figure 4.20: Number of iterations and wall clock time of the iterative velocity solvers with respect to the time step number.

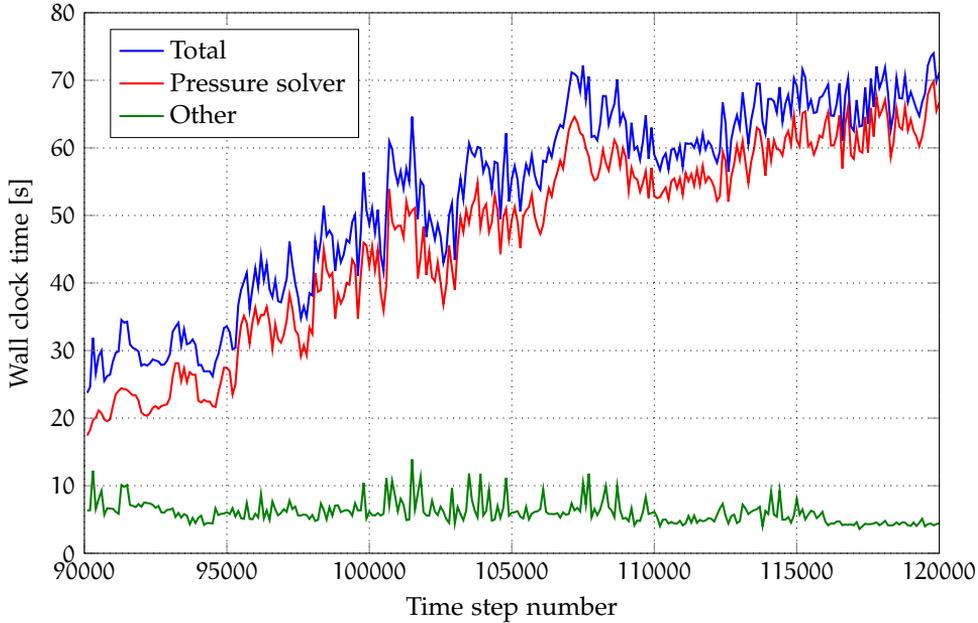
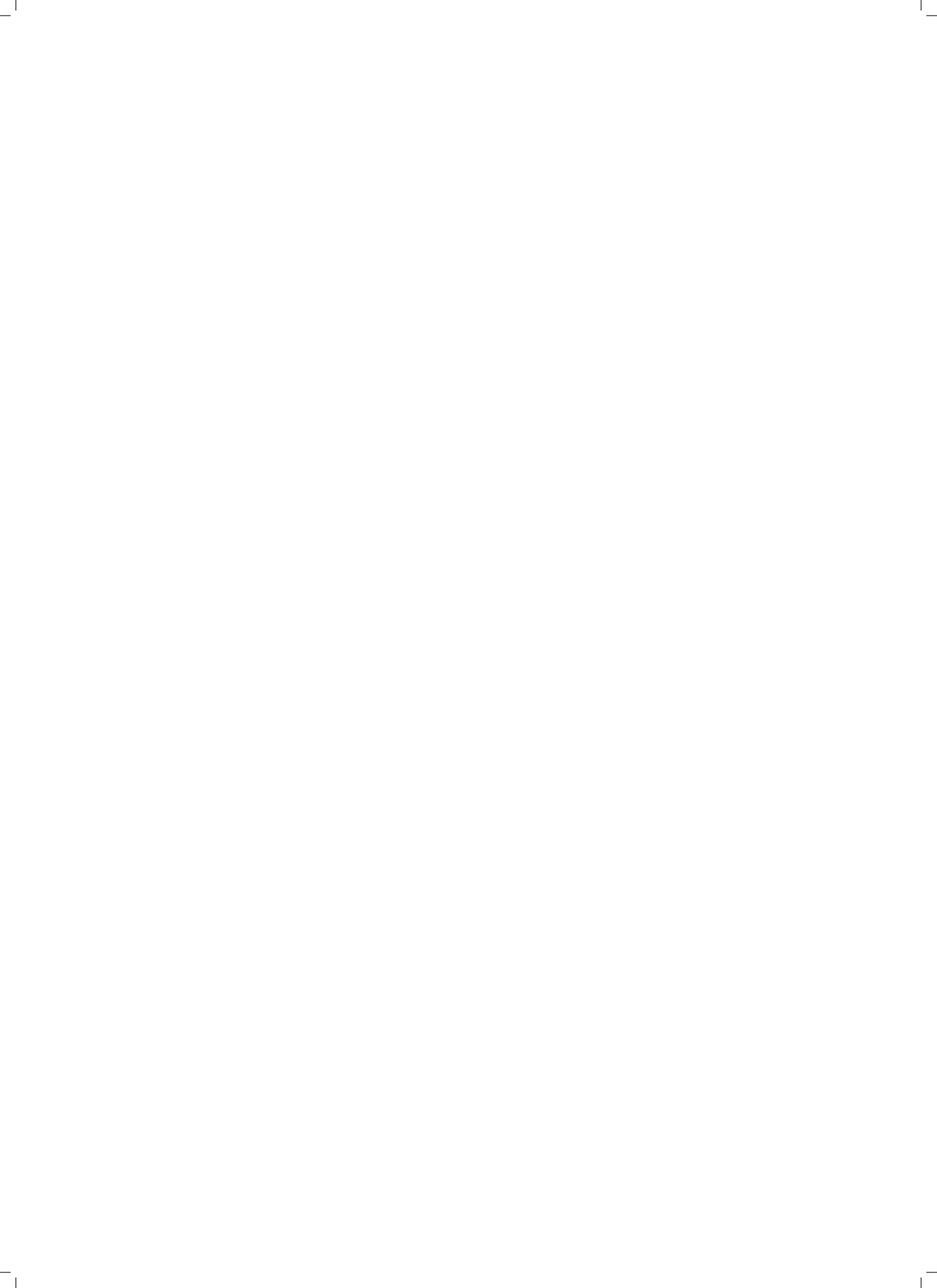


Figure 4.21: Scaled wall clock time with respect to the time step number.

4.6 CONCLUSIONS AND OUTLOOK

To investigate the usability of the multiple marker front-capturing method for its application to more challenging flows, it has been applied to a turbulent bubble-laden vertical channel downflow. Such a setup allows the study of the mutual interaction between the bubbles and the turbulent carrier fluid. A downflow ensures a bubble free wall region (bubbles tend to migrate towards the core of the channel in this flow configuration), which alleviates the absence of a bubble-wall interaction model. To be able to quantify the influence of the (coalescence/breakup of) bubbles on the turbulent flow, a staged strategy was chosen. First, as a reference a turbulent single-phase downflow (without any bubbles) has been simulated and analyzed. Although statistically not fully converged yet, preliminary results of the turbulent single-phase flow provide sufficient confidence that the computational method can be used to study turbulent channel flows. However, since the current method is designed to simulate two-phase flows, more efficient numerical methods are available for turbulent single-phase flows. Nevertheless, the presented single-phase simulations serve as an essential benchmark for subsequent two-phase flow simulations. Initially the flow was driven with a fixed bulk velocity, which after some time has been replaced by a fixed pressure gradient (which theoretically gives $Re_\tau = 180$). Next, at the time the single-phase flow approached statistical convergence, 384 bubbles were added to the channel. To enhance initial bubble interactions, bubbles were placed in a slightly perturbed regular array configuration.

Thanks to the master/slave technique (see Section 2.3.1) the computational cost of the advection of 384 bubbles is small, as depicted in Fig. 4.18. However, during the two-phase flow simulation some computational deficiencies were encountered. First, the time spend in the iterative pressure solver during the two-phase flow simulation is much higher (on average by more than a factor 4) than during the single-phase flow simulation. Due to an increase of the number of iterations required to reach a converged solution of the pressure Poisson equation, the time spend in the iterative pressure solver increases in time (during the last simulated time steps over a factor 6 more time is required per time step compared to single phase flow). To enhance the conditioning of the variable coefficient Poisson equation, that arises in solving the incompressible Navier-Stokes equations for two-phase flows, a deflation method is used (see Chapter 2). The increase of the number of iterations is possibly caused by a less effective deflation method (not all small eigenvalues are treated effectively). If this is indeed the case, a more dynamic choice of deflation vectors (based on interface positions) might lower the number of iterations [150]. Furthermore, recently it has been shown by Dodd and Ferrante [36] that the variable coefficient Poisson equation can be reduced to a constant coefficient equation, which can be solved with an FFT (Fast Fourier Transform)-based Poisson solver. Second, although the flow field is still divergence free, due to some non-physically high velocities inside some bubbles the time step becomes occasionally very small to satisfy the CFL time step criterion for numerical stability. The observed high velocities appear to originate from an inconsistency in the discretization of the advection of momentum and the advection of mass, which becomes apparent when the mass density ratio is very large. While the advective terms in the equations for the LS and VOF functions (mass) are discretized in conservative form, the advective terms in the momentum equations are discretized in non-conservative form. Consequently, small errors in the position of the interface (LS function) can result in spurious interfacial momentum transfer when the mass density ratio is high [33]. In recent literature a few remedies have been proposed to improve the coupling of the mass and momentum transport, which have shown to be effective and can form a starting point for future investigations. However, due to these accumulating computational issues (each time step becomes more expensive and more time steps are required) simulation of bubble-laden flows is currently not feasible within reasonable time with the DNS code developed in this project. Therefore only a limited number of dimensionless time scales has been computed, and the coalescence/breakup module has not been applied to the bubble-laden flow simulation at all. As a consequence only preliminary results were obtained, which can not be validated completely against existing numerical results. It can be concluded that the current multiple marker CLSVOF method is not optimal to study turbulent dispersed two-phase vertical channel flows in full detail. However, there are proven methods available to improve the efficiency of the pressure solver, and to achieve a consistent mass and momentum advection for bubble-laden flows. Therefore, after implementing these, it is expected that the multiple marker CLSVOF method will provide an accurate way to study a dispersion with many dispersed droplets including coalescence/breakup effects.



MAIN CONCLUSIONS AND OUTLOOK

In times of turbulence and rapid change, you must constantly be re-evaluating yourself relative to the new realities

— Brian Tracy

5.1 MAIN CONCLUSIONS

The objective of this work is to improve understanding of droplet-laden turbulent flows, in which the size of the droplets is much larger than the smallest turbulent length scale. In such flows coalescence and breakup events are likely to occur frequently, and need therefore to be modeled with sufficient accuracy. To this purpose, an accurate in-house method for the Direct Numerical Simulation (DNS) of interfacial flows has been developed further. The method is based on a front-capturing approach, which uses a fixed Cartesian grid for the computation of the fluid motion and two marker functions for the motion of each (deformable) interface. In this research three major milestones can be identified: efficiency improvements of the numerical method, the addition of coalescence and breakup modeling and large scale simulation of many dispersed fluid particles in a turbulent flow. To be more specific, in the latter case simulation and analysis of a bubbly downflow in a turbulent channel are performed. In this section the main results and conclusions are recapitulated.

5.1.1 *Efficiency improvements*

To be able to perform DNS of a large number of droplets, an accurate and efficient method is required. By using the multiple marker Coupled Level-Set and Volume-of-Fluid (CLSVOF) method [26, 27] sufficient accuracy is achieved. However, when the number of droplets is increased this implementation lacks efficiency. Since the focus of this research is on droplet-laden flows, the underlying causes for this lack of efficiency have been investigated in more detail. This research revealed two main causes: sequential advection of interfaces lead to an unbalanced load over the processing elements and the time spend in the pressure solver increases for a larger density ratio. The first issue is resolved by the implementation of a local multiple marker method based on a master/slave technique. This technique advects inter-

faces in parallel, leading to a load balanced algorithm and improved scaling with the number of interfaces. Where the global multiple marker method scales linearly with the amount of interfaces, the scaling for the local multiple marker method only depends on the maximum number of interfaces per processing unit. The second issue is alleviated by the addition of a second preconditioner to the preconditioned Conjugate Gradient (PCG) solver, based on the deflation technique [150]. The deflation method effectively removes unfavorable eigenvalues from the system to be solved. The resulting two-level preconditioned Conjugate Gradient (2L-PCG) solver decreases the required number of iterations by a factor of about 6, thereby reducing the overall time spend in the pressure solver by a factor of about 4. These results scale very well with increasing grid sizes, increasing number of interfaces, and a larger density ratio. Furthermore, to maximize parallelization benefits, the one-dimensional domain decomposition has been extended to a three-dimensional domain decomposition. A three-dimensional domain decomposition leads in general to smaller messages to be send. Overall, a speed up by a factor of about 8 is achieved for the buoyant rise of 48 droplets in a periodic box. Therefore, the improvements enable an accurate investigation of systems with a larger number of interfaces.

5.1.2 *Coalescence/breakup modeling*

Since coalescence and breakup events in droplet-laden flow are likely to occur, the numerical method has been extended with a coalescence and breakup module. A novel approach to determine the outcome of a binary droplet collision has been introduced, in which a coalescence model determines if and when two colliding droplets should coalesce. Although this idea has already been suggested by the group of Tryggvason [107, 155], the present work is the first in which it is applied within an interface resolving method. The coalescence model itself is based on a computationally efficient film drainage model [166], from which the film drainage time is determined. The coalescence model is incorporated into the efficient multiple marker CLSVOF method, which is capable of accurately simulating $\mathcal{O}(10^3)$ dispersed droplets. Although its incorporation within an interface resolving method is laborious, it is considered inevitable for study of dense droplet-laden flows where droplets will frequently coalesce. Moreover, since the implementation of the coalescence model in the CLSVOF method is generic, incorporation of coalescence models other than the currently used film drainage model is straightforward. Breakup will numerically occur when the marker functions can not resolve a part of the interface anymore. Since droplet breakup is a fast process, the effect of numerical breakup on the collision dynamics is negligible. However, the detection of breakup and the corresponding creation of new marker functions is essential to ensure physical coalescence events at later stages of the simulation. Furthermore, creation of additional local marker functions allows an continuous load balancing over available resources.

To validate the extension of the CLSVOF method with the coalescence/breakup modules, simulations of (near) head-on binary droplet collisions have been per-

formed. First, experimentally observed film drainage times have been used to determine the instant of coalescence. The results of these binary droplet collisions show that, if a correct film drainage time is available, the CLSVOF method can reproduce existing experimental and numerical results. This implies that the droplets are rather insensitive to the details of the simulated flow within the thin film, since these effects are not fully resolved. Next, to study the influence of the film drainage time on the collision dynamics, several film drainage times for a specific collision have been analyzed. As expected, results show that the collision dynamics strongly depends on the film drainage time. Therefore, the coalescence model should provide accurate film drainage times to ensure realistic collision dynamics. Although the order of magnitude of the predicted film drainage times by the current model is correct, quantitatively it is not sufficiently accurate. Potential causes for this deviation are the empirical parameter δ (which has no clear physical meaning), the truncated-sphere approximation of the colliding droplets, and the assumption of axisymmetric film drainage (recent experiments have shown asymmetric drainage [111]). However, due to the complexity of the film drainage problem it may not be straightforward to develop more accurate film drainage models with similar computational efficiency. Finally, numerical results show that the effect of grid resolution on the coalescence process is limited. Only the instant of first contact is slightly grid dependent, but this effect can easily be circumvented. Therefore, this aspect should be considered less important than the accuracy of the film drainage model, which has by far more influence on the collision dynamics. However, if the amount of deformation is large (with respect to the grid resolution), the computed interface curvature becomes less accurate. These errors in curvature directly influence surface tension forces, and thereby significantly affect the interface shape over time.

5.1.3 Large scale simulations

The large scale simulations performed in this work are bubbly downflows in a turbulent channel, where the continuous and dispersed phases are water and air, respectively. This setup allows the study of the mutual interaction between the bubbles and the turbulent carrier fluid, which properties can be validated by available results. Furthermore, a downflow ensures a bubble free wall region (bubbles tend to migrate towards the core of the channel), which alleviates the absence of a bubble-wall interaction model. Before this simulation could be performed, first a turbulent single-phase downflow (without any bubbles) with a friction Reynolds number of $Re_\tau = 180$ (which corresponds to a bulk Reynolds number of $Re_b = \frac{U_b 2H}{\nu_c} \approx 5534$) has been simulated and analyzed. Although this initial simulation showed a slow statistical convergence (the current method is optimized to simulate two-phase flows, more efficient numerical methods are available for turbulent single-phase flows), obtained results provided sufficient confidence for the quality of the code. Next, 384 finite-size air bubbles (with a diameter of 10% of the channel width) were added to the channel, and placed in a slightly perturbed regular array configuration to enhance initial bubble interactions. Thanks to the master/slave technique,

the computational cost of the advection of 384 bubbles is small. To investigate the feasibility of the code to perform these large two-phase flow simulations, the coalescence/breakup of bubbles has been disabled. Although the simulations started promising, during the two-phase flow simulation some computational deficiencies were encountered.

First, due to an increase of the number of iterations required to reach the convergence criterion, the time spend in the iterative pressure solver increases in time (during the last simulated time steps over a factor 6 more time is required per time step compared to single phase flow). To enhance the conditioning of the variable coefficient Poisson equation, that arises in solving the incompressible Navier-Stokes equations for two-phase flows, a deflation method is used (see Chapter 2). The increase of the number of iterations is possibly caused by a less effective deflation method (not all small eigenvalues are treated effectively). If this is indeed the case, a more dynamic choice of deflation vectors (based on interface positions) might lower the number of iterations [150]. Second, although the flow field is still divergence free, due to non-physically high velocities inside some bubbles the time step becomes occasionally very small to satisfy the CFL time step criterion for numerical stability. The observed high velocities appear to originate from an inconsistency in the discretization of the advection of momentum and the advection of mass around the interface, which becomes apparent when the mass density ratio is large. While the advective terms in the equations for the LS and VOF functions (mass) are discretized in conservative form, the advective term in the momentum equation is discretized in non-conservative form. Consequently, small errors in the position of the interface (LS function) can result in spurious interfacial momentum transfer when the mass density ratio is high [33]. In recent literature [33, 123, 125] a few remedies have been proposed to improve the coupling of the mass and momentum transport, which have shown to be effective and can form a starting point for future investigations. However, due to these accumulating computational issues (each time step becomes more expensive and more time steps are required), a complete analysis of the bubbly downflow is with the current setup not feasible within reasonable time. Therefore only a limited number of dimensionless time scales have been computed, and the coalescence/breakup module has not been applied to the bubble-laden flow simulation at all. As a consequence only preliminary results were obtained, which can not be validated completely against existing numerical results. It can be concluded that the current multiple marker CLSVOF method is not optimal to study the chosen turbulent dispersed two-phase vertical channel flow in full detail. However, there are options available to improve the efficiency of the pressure solver, and to achieve a consistent mass and momentum advection for bubble-laden flows. Therefore, after implementing these or setting up a simulation with less ambitious parameters, it is expected that the presented multiple marker CLSVOF method will be capable of studying turbulent dense dispersed two-phase flows including coalescence and breakup.

5.2 OUTLOOK

As the main results in the previous section show, important advances have been made for the numerical modeling of turbulent dense dispersed two-phase flows. The method as presented here is capable of determining collision frequencies and coalescence efficiencies in droplet-laden flows. Nevertheless, there are still some issues that require attention for future improvement. These issues can be categorized in four major groups: numerical efficiency, numerical accuracy, physical realism and validation.

1. **Numerical efficiency**

Although some optimization of the deflation method might be possible, the iterative pressure solver remains a major bottleneck in the entire method. Fortunately, between the time after finishing this research and the actual publication of this thesis, Dodd and Ferrante [36] showed that the variable coefficient Poisson equation can be reduced to a constant coefficient equation. The advantage of this approach is that constant coefficient equations can be solved by an FFT (Fast Fourier Transform)-based Poisson solver. This innovation would significantly decrease the computational time, and thereby allow the simulation of more challenging problems.

2. **Numerical accuracy**

Some implementations of the numerical method depend strongly on the grid resolution, and therefore influence the numerical accuracy. For example, in regions with a large amount of deformation (with respect to the grid size) the current Height Function (HF) implementation becomes inaccurate and provides a wrong estimate of the local curvature. As shown in Section 3.4, over time these errors deform the interface position in a non-physical way. Possible remedies against issues like this are to make the implementation itself less grid dependent [81, 93, 95], or employ Adaptive Mesh Refinement (AMR) techniques (overlapping grids, tree-based grids). Here the work of Cervone et al. [20] might be of interest, who applied a multilevel VOF method with several levels of refinement. Furthermore, for two-phase moderate Reynolds number flows the regularization of the viscosity reduces the spatial convergence from second to first order when the viscosity ratio is not equal to unity [26]. Finally, as shown in Section 4.5, discretization inconsistencies in the discretization schemes for advective mass and momentum transport can give rise to spurious errors in the velocity field, especially in the presence of strong shear at interfaces and high density ratios [33]. Knowledge of the limitations of numerical methods is essential for the confidence in the obtained results with those methods.

3. **Physical realism**

The amount of physical realism determines the applicability of the numerical method. As described in Chapter 4, a downflow was chosen to ensure a bubble free wall region to alleviate the absence of bubble-wall interaction

modeling. If the numerical method would be able to simulate such events with physical realism, also upflow simulations would be possible. Besides additions like this, also mass- and heat transfer between the two phases would broaden the field of application. Here the work of Talebanfard and Boersma [149] should be mentioned, who applied the CLSVOF method to model heat transfer in colliding droplets in merging and bouncing regimes. Finally, further advancements in the the area of variable interfacial tension (Marangoni effects) would also be of general interest.

Thanks to the master/slave technique, interfacial procedures (like advection, reinitialization, computation of local viscosity, density, and surface tension fields) are entirely handled by a single processing element (PE). This aspect considerably simplifies all subsequent extensions of the method which depend on interfacial properties, since no parallelization within such an extension is required.

4. Validation

Finally, validation of the applied methods and software is essential in building confidence in obtained results and show their robustness. Besides earlier research of Coyajee [26, 27] and the research presented in this thesis, Vreeken [159] and Jansen [63] have recently successfully applied the current method and software to investigate rising droplets in a periodic domain and oblique droplet collisions, respectively. The latter validated the assumption made in Chapter 3 that an oblique droplet collision can be treated as a head-on droplet collision if the relative velocity between the two impacting droplets is equal. Possible directions for future validation studies are the quality of the film drainage model (which needs to be combined with accurate measurements of film drainage times with various initial parameters), and the ability of the current method to simulate unequal sized binary collisions (here the equivalent droplet diameter [23] might be of use) and droplet-wall collisions. In a broader sense, simulations to investigate the mutual influence between the dispersed phase and the turbulence remain of key interest.

BIBLIOGRAPHY

- [1] ADALSTEINSSON, D. AND SETHIAN, J. A. A Fast Level Set Method for Propagating Interfaces. *Journal of Computational Physics*, 118(2):269–277, 1995.
- [2] ADAM, J. R., LINDBLAD, N. R., AND HENDRICKS, C. D. The Collision, Coalescence, and Disruption of Water Droplets. *Journal of Applied Physics*, 39(11):5173, 1968.
- [3] DEL ÁLAMO, J. C. AND JIMÉNEZ, J. Spectra of the very large anisotropic scales in turbulent channels. *Physics of Fluids*, 15(6):L41–L44, 2003.
- [4] ASHGRIZ, N. AND GIVI, P. Binary collision dynamics of fuel droplets. *International Journal of Heat and Fluid Flow*, 8(3):205–210, 1987.
- [5] AULISA, E., MANSERVISI, S., AND SCARDOVELLI, R. A mixed markers and volume-of-fluid method for the reconstruction and advection of interfaces in two-phase and free-boundary flows. *Journal of Computational Physics*, 188(2):611–639, 2003.
- [6] BALACHANDAR, S. AND EATON, J. K. Turbulent Dispersed Multiphase Flow. *Annu. Rev. Fluid Mech*, 42:111–133, 2010.
- [7] BERMAN, A. AND PLEMMONS, R. J. *Nonnegative matrices in the mathematical sciences*. SIAM, 1994.
- [8] BOLOTNOV, I. A., JANSEN, K. E., DREW, D. A., OBERAI, A. A., LAHEY JR., R. T., AND PODOWSKI, M. Z. Detached Direct Numerical Simulations of turbulent two-phase bubbly channel flow. *International Journal of Multiphase Flow*, 37(6):647–659, 2011.
- [9] BRACKBILL, J. U., KOTHE, D. B., AND ZEMACH, C. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335–354, 1992.
- [10] BRANDT, T. Direct Numerical Simulation of Turbulent Channel Flow using Finite Difference Scheme. In *VIII Finnish Mechanics Days*, pages 639–650, 2003.
- [11] BRAZIER-SMITH, P. R., JENNINGS, S. G., AND LATHAM, J. The Interaction of Falling Water Drops: Coalescence. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 326(1566):393–408, 1972.
- [12] BRENN, G. AND FROHN, A. Collision and merging of two equal droplets of propanol. *Experiments in Fluids*, 7(7):441–446, 1989.
- [13] BRIDSON, R. E. *Computational aspects of dynamic surfaces*. PhD thesis, Stanford University, 2003.

- [14] BUNNER, B. AND TRYGGVASON, G. Direct Numerical Simulations of three-dimensional bubbly flows. *Physics of Fluids*, 11(8):1967–1969, 1999.
- [15] BUNNER, B. AND TRYGGVASON, G. An Examination of the Flow Induced by the Motion of Many Buoyant Bubbles. *J. Vis.*, 2(2):153–158, 1999.
- [16] BUNNER, B. AND TRYGGVASON, G. Dynamics of Homogeneous Bubbly Flows Part 1. Rise Velocity and Microstructure of the Bubbles. *Journal of Fluid Mechanics*, 466:17–52, 2002.
- [17] BUNNER, B. AND TRYGGVASON, G. Dynamics of Homogeneous Bubbly Flows Part 2. Velocity Fluctuations. *Journal of Fluid Mechanics*, 466:53–84, 2002.
- [18] BUNNER, B. AND TRYGGVASON, G. Effect of Bubble Deformation on the Properties of Bubbly Flows. *Journal of Fluid Mechanics*, 495:77–118, 2003.
- [19] BURRILL, K. A. AND WOODS, D. R. Film shapes for deformable drops at liquid-liquid interfaces. III. Drop rest-times. *Journal of Colloid and Interface Science*, 42(1):35–51, 1973.
- [20] CERVONE, A., MANSERVISI, S., AND SCARDOVELLI, R. An optimal constrained approach for divergence-free velocity interpolation and multilevel VOF method. *Computers & Fluids*, 47(1):101–114, 2011.
- [21] CHANG, Y. C., HOU, T. Y., MERRIMAN, B., AND OSHER, S. A Level Set Formulation of Eulerian Interface Capturing Methods for Incompressible Fluid Flows. *Journal of Computational Physics*, 124(2):449–464, 1996.
- [22] CHESTERS, A. K. The modelling of coalescence processes in fluid-liquid dispersions : a review of current understanding. *Chemical engineering research & design*, 69A:259–270, 1991.
- [23] CHESTERS, A. K. AND HOFMAN, G. Bubble coalescence in pure liquids. *Applied Scientific Research*, 38(1):353–361, 1982.
- [24] CLIFT, R., GRACE, J., AND WEBER, M. *Bubbles, Drops, and Particles*. Academic Press, Inc., London, United Kingdom, 1978.
- [25] CORBETT, R. D. *Point-based level sets and progress towards unorganized particle based fluids*. PhD thesis, The University of British Columbia, 2005.
- [26] COYAJEE, E. R. A. *A front-capturing method for the numerical simulation of dispersed two-phase flow*. PhD thesis, TU Delft, 2007.
- [27] COYAJEE, E. R. A. AND BOERSMA, B. J. Numerical simulation of drop impact on a liquid-liquid interface with a multiple marker front-capturing method. *Journal of Computational Physics*, 228(12):4444–4467, 2009.
- [28] CROWE, C. T., SCHWARZKOPF, J. D., SOMMERFELD, M., AND TSUJI, Y. *Multiphase Flows with Droplets and Particles*. CRC Press, 2 edition, 2011.

- [29] DAVENPORT, J., HUGHES, R. N., SHORTEN, M., AND LARSEN, P. S. Drag reduction by air release promotes fast ascent in jumping emperor penguins – a novel hypothesis. *Marine Ecology Progress Series*, 430:171–182, 2011.
- [30] DAVIS, R. H., SCHONBERG, J. A., AND RALLISON, J. M. The lubrication force between two viscous drops. *Physics of Fluids A: Fluid Dynamics*, 1(1):77–81, 1989.
- [31] DEAN, R. B. Reynolds Number Dependence of Skin Friction and Other Bulk Flow Variables in Two-Dimensional Rectangular Duct Flow. *Journal of Fluids Engineering*, 100(2):215, 1978.
- [32] DEEN, N. G., VAN SINT ANNALAND, M., AND KUIPERS, J. A. M. Multi-scale modeling of dispersed gasliquid two-phase flow. *Chemical Engineering Science*, 59(89):1853–1861, 2004.
- [33] DESJARDINS, O. AND MOUREAU, V. Methods for multiphase flows with high density ratio. In *Proceedings of the Summer Program 2010*, pages 313–322, 2010.
- [34] DESJARDINS, O. AND PITSCHE, H. A spectrally refined interface approach for simulating multiphase flows. *Journal of Computational Physics*, 228(5):1658–1677, 2009.
- [35] DIJKHUIZEN, W., ROGHAI, I., ANNALAND, M. V. S., AND KUIPERS, J. DNS of gas bubbles behaviour using an improved 3d front tracking model–Model development. *Chemical Engineering Science*, 65(4):1427–1437, 2010.
- [36] DODD, M. S. AND FERRANTE, A. A fast pressure-correction method for incompressible two-fluid flows. *Journal of Computational Physics*, 273:416–434, 2014.
- [37] DOSTÁL, Z. Conjugate gradient method with preconditioning by projector. *International Journal of Computer Mathematics*, 23(3):315–323, 1988.
- [38] ELGHOBASHI, S. On predicting particle-laden turbulent flows. *Applied Scientific Research*, 52(4):309–329, 1994.
- [39] ENRIGHT, D., FEDKIW, R., FERZIGER, J., AND MITCHELL, I. A Hybrid Particle Level Set Method for Improved Interface Capturing. *J. Comput. Phys*, 183: 83–116, 2002.
- [40] ESMAEELI, A. AND TRYGGVASON, G. An Inverse Energy Cascade in Two-Dimensional Low Reynolds Number Bubbly Flows. *Journal of Fluid Mechanics Digital Archive*, 314:315–330, 1996.
- [41] ESMAEELI, A. AND TRYGGVASON, G. Direct Numerical Simulations of Bubbly Flows Part 1. Low Reynolds Number Arrays. *Journal of Fluid Mechanics*, 377: 313–345, 1998.
- [42] ESMAEELI, A. AND TRYGGVASON, G. Direct Numerical Simulations of Bubbly Flows Part 2. Moderate Reynolds Number Arrays. *Journal of Fluid Mechanics*, 385:325–358, 1999.

- [43] ESMAEELI, A. AND TRYGGVASON, G. A Direct Numerical Simulation study of the buoyant rise of bubbles at $O(100)$ Reynolds number. *Physics of Fluids*, 17 (9):093303, 2005.
- [44] FEDKIW, R. P. AND LIU, X.-D. The Ghost Fluid Method for Viscous Flows, 1998.
- [45] FOOTE, G. B. The Water Drop Rebound Problem: Dynamics of Collision. *Journal of the Atmospheric Sciences*, 32(2):390–402, 1975.
- [46] FRANCOIS, M. M., CUMMINS, S. J., DENDY, E. D., KOTHE, D. B., SICILIAN, J. M., AND WILLIAMS, M. W. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *Journal of Computational Physics*, 213(1):141–173, 2006.
- [47] GAUDLITZ, D. AND ADAMS, N. On Implementing the Hybrid Particle-Level-Set Method on Supercomputers for Two-Phase Flow Simulations. In *High Performance Computing in Science and Engineering '07*, pages 445–456. Springer Berlin Heidelberg, 2008.
- [48] GAUDLITZ, D. AND ADAMS, N. A. A Hybrid Particle Level Set Method for Two-Phase Flow. *PAMM*, 4(1):506–507, 2004.
- [49] GHODS, S. AND HERRMANN, M. A consistent rescaled momentum transport method for simulating large density ratio incompressible multiphase flows using level set methods. *Physica Scripta*, 2013(T155):014050, 2013.
- [50] GHOSH, S., DÁVILA, J., HUNT, J., SRDIC, A., FERNANDO, H., AND JONAS, P. How turbulence enhances coalescence of settling particles with applications to rain in clouds. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 461(2062):3059–3088, 2005.
- [51] GOLUB, G. H. AND LOAN, C. F. V. *Matrix computations*. Johns Hopkins University Press, 1996.
- [52] GRACE, J., WAIREGI, T., AND NGUYEN, T. Shapes and velocities of single drops and bubbles moving freely Through Immiscible Liquids. *Chemical Engineering Research and Design*, 54a:167 – 173, 1976.
- [53] HARLEMAN, M. J. W. *On the effect of turbulence on bubbles: experiments and numerical simulations of bubbles in wall-bounded flows*. PhD thesis, TU Delft, 2012.
- [54] HARLOW, F. H. AND WELCH, J. E. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *Physics of Fluids*, 8 (12):2182–2189, 1965.
- [55] HELMSEN, J., COLELLA, P., AND PUCKETT, E. Non-convex profile evolution in two dimensions using volume of fluids. Technical Report LBNL-40693, Lawrence Berkeley National Laboratory, 1997.

- [56] HERRMANN, M. A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *Journal of Computational Physics*, 227(4):2674–2706, 2008.
- [57] HERRMANN, M. A parallel Eulerian interface tracking/Lagrangian point particle multi-scale coupling procedure. *Journal of Computational Physics*, 229(3):745–759, 2010.
- [58] HEWITT, G. F. AND HALL-TAYLOR, N. S. *Annular two-phase flow*. Pergamon Press, 1970.
- [59] HIRT, C. W. AND NICHOLS, B. D. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225, 1981.
- [60] HOITINGA, W. *Goal Adaptive Discretization of a One-Dimensional Boltzmann Equation*. PhD thesis, TU Delft, 2011.
- [61] HOUSTON, B., NIELSEN, M. B., BATTY, C., NILSSON, O., AND MUSETH, K. Hierarchical RLE level set: A compact and versatile deformable surface representation. *ACM Trans. Graph.*, 25(1):151–175, 2006.
- [62] HOWARTH, W. J. Coalescence of drops in a turbulent flow field. *Chemical Engineering Science*, 19(1):33–38, 1964.
- [63] JANSEN, Y. L. Oblique droplet collisions - A Numerical Investigation. MSc Thesis, Delft University of Technology, 2015.
- [64] JIANG, X. AND JAMES, A. Numerical simulation of the head-on collision of two equal-sized drops with van der Waals forces. *Journal of Engineering Mathematics*, 59(1):99–121, 2007.
- [65] JIANG, Y. J., UMEMURA, A., AND LAW, C. K. An Experimental Investigation on the Collision Behaviour of Hydrocarbon Droplets. *Journal of Fluid Mechanics*, 234:171–190, 1992.
- [66] JIMÉNEZ, J. AND MOIN, P. The minimal flow unit in near-wall turbulence. *Journal of Fluid Mechanics*, 225:213, 1991.
- [67] KAASSCHIETER, E. F. Preconditioned conjugate gradients for solving singular systems. *Journal of Computational and Applied Mathematics*, 24(1-2):265–275, 1988.
- [68] KAN, J. v. A Second-Order Accurate Pressure-Correction Scheme for Viscous Incompressible Flow. *SIAM Journal on Scientific and Statistical Computing*, 7(3): 870–891, 1986.
- [69] KANG, M., FEDKIW, R. P., AND LIU, X.-D. A Boundary Condition Capturing Method for Multiphase Incompressible Flow. *Journal of Scientific Computing*, 15(3):323–360, 2000.

- [70] KIM, J., MOIN, P., AND MOSER, R. Turbulence Statistics in Fully Developed Channel Flow at Low Reynolds Number. *Journal of Fluid Mechanics*, 177:133–166, 1987.
- [71] KLASEBOER, E., CHEVAILLIER, J. P., GOURDON, C., AND MASBERNAT, O. Film Drainage between Colliding Drops at Constant Approach Velocity: Experiments and Modeling. *Journal of Colloid and Interface Science*, 229(1):274–285, 2000.
- [72] KLEEFMAN, K. M. T. *Water impact loading on offshore structures - A numerical study*. PhD thesis, University of Groningen, 2005.
- [73] KNOLL, D. A. AND RIDER, W. J. A Multigrid Preconditioned Newton–Krylov Method. *SIAM Journal on Scientific Computing*, 21:691–710, 1999.
- [74] KOLMOGOROV, A. The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds’ Numbers. *Akademiia Nauk SSSR Doklady*, 30: 301–305, 1941.
- [75] KWAKKEL, M., BREUGEM, W.-P., AND BOERSMA, B. J. An efficient multiple marker front-capturing method for two-phase flows. *Computers & Fluids*, 63: 47–56, 2012.
- [76] KWAKKEL, M., BREUGEM, W.-P., AND BOERSMA, B. J. Extension of a CLSVOF method for droplet-laden flows with a coalescence/breakup model. *Journal of Computational Physics*, 253:166–188, 2013.
- [77] LAFAURIE, B., NARDONE, C., SCARDOVELLI, R., ZALESKI, S., AND ZANETTI, G. Modelling Merging and Fragmentation in Multiphase Flows with SURFER. *Journal of Computational Physics*, 113(1):134–147, 1994.
- [78] LEHR, F. AND MEWES, D. A transport equation for the interfacial area density applied to bubble columns. *Chemical Engineering Science*, 56(3):1159–1166, 2001.
- [79] LI, X. AND FRITSCHING, U. Numerical investigation of binary droplet collisions in all relevant collision regimes. *The Journal of Computational Multiphase Flows*, 3(4):207–224, 2011.
- [80] LIAO, Y. AND LUCAS, D. A literature review on mechanisms and models for the coalescence process of fluid particles. *Chemical Engineering Science*, 65(10), 2010.
- [81] LIOVIC, P., FRANCOIS, M., RUDMAN, M., AND MANASSEH, R. Efficient simulation of surface tension-dominated flows through enhanced interface geometry interrogation. *Journal of Computational Physics*, 229(19):7520–7544, 2010.
- [82] LIU, X.-D., FEDKIW, R. P., AND KANG, M. A Boundary Condition Capturing Method for Poisson’s Equation on Irregular Domains. *Journal of Computational Physics*, 160(1):151–178, 2000.

- [83] LOEWENBERG, M. AND HINCH, E. J. Numerical Simulation of a Concentrated Emulsion in Shear Flow. *Journal of Fluid Mechanics*, 321:395–419, 1996.
- [84] LOEWENBERG, M. AND HINCH, E. J. Collision of Two Deformable Drops in Shear Flow. *Journal of Fluid Mechanics*, 338:299–315, 1997.
- [85] LOSASSO, F., GIBOU, F., AND FEDKIW, R. Simulating water and smoke with an octree data structure. In *ACM SIGGRAPH 2004 Papers*, pages 457–462, Los Angeles, California, 2004. ACM.
- [86] LOTH, E. *Particles, drops and bubbles: fluid dynamics and numerical methods*. Cambridge University Press, 2011. In preparation.
- [87] LU, E. J.-L. AND OKUNBOR, D. I. A massively parallel fast multipole algorithm in three dimensions. In *Proceedings of 5th IEEE International Symposium on High Performance Distributed Computing*, 1996, pages 40–48. IEEE, 1996.
- [88] LU, J. AND TRYGGVASON, G. Numerical study of turbulent bubbly downflows in a vertical channel. *Physics of Fluids*, 18(10):103302, 2006.
- [89] LU, J. AND TRYGGVASON, G. Effect of bubble size in turbulent bubbly downflow in a vertical channel. *Chemical Engineering Science*, 62(11):3008–3018, 2007.
- [90] LU, J. AND TRYGGVASON, G. Effect of bubble deformability in turbulent bubbly upflow in a vertical channel. *Physics of Fluids*, 20(4):040701, 2008.
- [91] LU, J., FERNÁNDEZ, A., AND TRYGGVASON, G. The effect of bubbles on the wall drag in a turbulent channel flow. *Physics of Fluids*, 17(9):095102, 2005.
- [92] LU, J., BISWAS, S., AND TRYGGVASON, G. A DNS study of laminar bubbly flows in a vertical channel. *International Journal of Multiphase Flow*, 32(6):643–660, 2006.
- [93] LÓPEZ, J. AND HERNÁNDEZ, J. On reducing interface curvature computation errors in the height function technique. *Journal of Computational Physics*, 229(13):4855–4868, 2010.
- [94] LÓPEZ, J., HERNÁNDEZ, J., GÓMEZ, P., AND FAURA, F. An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows. *Journal of Computational Physics*, 208(1):51–74, 2005.
- [95] LÓPEZ, J., ZANZI, C., GÓMEZ, P., ZAMORA, R., FAURA, F., AND HERNÁNDEZ, J. An improved height function technique for computing interface curvature from volume fractions. *Computer Methods in Applied Mechanics and Engineering*, 198(33-36):2555–2564, 2009.
- [96] LÖRSTAD, D., FRANCOIS, M., SHYY, W., AND FUCHS, L. Assessment of volume of fluid and immersed boundary methods for droplet computations. *International Journal for Numerical Methods in Fluids*, 46(2):109–125, 2004.

- [97] MACLACHLAN, S. P., TANG, J. M., AND VUIK, C. Fast and robust solvers for pressure-correction in bubbly flow problems. *J. Comput. Phys.*, 227(23):9742–9761, 2008.
- [98] MARTINEZ, J., CHEHATA, D., VAN GILS, D. P. M., SUN, C., AND LOHSE, D. On bubble clustering and energy spectra in pseudo-turbulence. *0912.1620*, 2009.
- [99] MENARD, T., TANGUY, S., AND BERLEMONT, A. Coupling level set/VOF/ghost fluid methods: Validation and application to 3d simulation of the primary break-up of a liquid jet. *International Journal of Multiphase Flow*, 33(5):510–524, 2007.
- [100] MOIN, P. AND MAHESH, K. Direct Numerical Simulation: A Tool in Turbulence Research. *Annu. Rev. Fluid Mech.*, 30:539–578, 1998.
- [101] MOSER, R. D., KIM, J., AND MANSOUR, N. N. Direct Numerical simulation of turbulent channel flow up to $Re_{\tau} = 590$. *Physics of Fluids*, 11(4):943–945, 1999.
- [102] MULDER, W., OSHER, S., AND SETHIAN, J. A. Computing interface motion in compressible gas dynamics. *Journal of Computational Physics*, 100(2):209–228, 1992.
- [103] NICOLAIDES, R. A. Deflation of Conjugate Gradients with Applications to Boundary Value Problems. *SIAM Journal on Numerical Analysis*, 24(2):355–365, 1987.
- [104] NIELSEN, M. AND MUSETH, K. Dynamic Tubular Grid: An Efficient Data Structure and Algorithms for High Resolution Level Sets. *Journal of Scientific Computing*, 26(3):261–299, 2006.
- [105] NIKOLOPOULOS, N., NIKAS, K.-S., AND BERGELES, G. A numerical investigation of central binary collision of droplets. *Computers & Fluids*, 38(6):1191–1202, 2009.
- [106] NIKOLOPOULOS, N., STROTOS, G., NIKAS, K., AND BERGELES, G. The effect of Weber number on the central binary collision outcome between unequal-sized droplets. *International Journal of Heat and Mass Transfer*, 55(78):2137–2150, 2012.
- [107] NOBARI, M. R., JAN, Y.-J., AND TRYGGVASON, G. Head-on collision of drops—A numerical investigation. *Physics of Fluids*, 8(1):29–42, 1996.
- [108] NOBARI, M. R. H. *Numerical simulation of oscillation, collision, and coalescence of drops*. PhD thesis, University of Michigan, 1993. Ph.D.
- [109] NOBARI, M. R. H. AND TRYGGVASON, G. Numerical Simulations of Three-Dimensional Drop Collisions. *AIAA journal*, 1996.
- [110] NOH, W. AND WOODWARD, P. SLIC (Simple Line Interface Calculation). In van de Vooren, A. and Zandbergen, P., editors, *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28 July 2, 1976*

- Twente University, Enschede*, volume 59 of *Lecture Notes in Physics*, pages 330–340. Springer Berlin / Heidelberg, 1976.
- [111] OLDENZIEN, G., DELFOS, R., AND WESTERWEEL, J. Measurements of liquid film thickness for a droplet at a two-fluid interface. *Physics of Fluids*, 24(2):022106–022106–18, 2012.
 - [112] OSHER, S. AND FEDKIW, R. P. Level Set Methods: An Overview and Some Recent Results. *Journal of Computational Physics*, 169(2):463–502, 2001.
 - [113] OSHER, S. AND FEDKIW, R. P. *Level set methods and dynamic implicit surfaces*. Springer, 2003.
 - [114] OSHER, S. AND SETHIAN, J. A. Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.
 - [115] PAN, K.-L., LAW, C. K., AND ZHOU, B. Experimental and mechanistic description of merging and bouncing in head-on binary droplet collision. *Journal of Applied Physics*, 103(6):064901, 2008.
 - [116] PAN, Y. AND SUGA, K. Numerical simulation of binary liquid droplet collision. *Physics of Fluids*, 17(8):082105–14, 2005.
 - [117] VAN DER PIJL, S. P. *Computation of bubbly flows with a mass-conserving level-set method*. PhD thesis, TU Delft, 2005.
 - [118] PILCH, M. AND ERDMAN, C. Use of breakup time data and velocity history data to predict the maximum size of stable fragments for acceleration-induced breakup of a liquid drop. *International Journal of Multiphase Flow*, 13(6):741–757, 1987.
 - [119] PRAKASH, V. N., TAGAWA, Y., CALZAVARINI, E., MERCADO, J. M., TOSCHI, F., LOHSE, D., AND SUN, C. How gravity and size affect the acceleration statistics of bubbles in turbulence. *New Journal of Physics*, 14(10):105017, 2012.
 - [120] PUCKETT, E. G., ALMGREN, A. S., BELL, J. B., MARCUS, D. L., AND RIDER, W. J. A High-Order Projection Method for Tracking Fluid Interfaces in Variable Density Incompressible Flows. *Journal of Computational Physics*, 130(2):269–282, 1997.
 - [121] QIAN, J. AND LAW, C. K. Regimes of Coalescence and Separation in Droplet Collision. *Journal of Fluid Mechanics*, 331:59–80, 1997.
 - [122] RADHAKRISHNANA, K. AND HINDMARSH, A. Description and use of LSODE, the Livermore solver for ordinary differential equations. LLNL Report UCRL-ID-113855, LLNL Report, 1993.
 - [123] RAESSI, M. AND PITSCH, H. Consistent mass and momentum transport for simulating incompressible interfacial flows with large density ratios using the level set method. *Computers & Fluids*, 63:70–81, 2012.

BIBLIOGRAPHY

- [124] RIDER, W. J. AND KOTHE, D. B. Reconstructing Volume Tracking. *Journal of Computational Physics*, 141:112–152, 1998.
- [125] RUDMAN, M. A volume-tracking method for incompressible multifluid flows with large density variations. *International Journal for Numerical Methods in Fluids*, 28:357–378, 1998.
- [126] SABONI, A., GOURDON, C., AND CHESTERS, A. K. Drainage and Rupture of Partially Mobile Films during Coalescence in Liquid-Liquid Systems under a Constant Interaction Force. *Journal of Colloid and Interface Science*, 175(1):27–35, 1995.
- [127] SCARDOVELLI, R. AND ZALESKI, S. Direct Numerical Simulation of Free-Surface and Interfacial flow. *Annual Review of Fluid Mechanics*, 31:567–603, 1999.
- [128] SCHLICHTING, H. AND GERSTEN, K. *Boundary-Layer Theory*. Springer Science & Business Media, 8th edition, 2000.
- [129] SETHIAN, J. A. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 2 edition, 1999.
- [130] SETHIAN, J. A. Evolution, Implementation, and Application of Level Set and Fast Marching Methods for Advancing Fronts. *Journal of Computational Physics*, 169(2):503–555, 2001.
- [131] SHAW, R. A. Particle-Turbulence Interactions in Atmospheric Clouds. *Annual Review of Fluid Mechanics*, 35(1):183–227, 2003.
- [132] SHIN, S. AND JURIC, D. Modeling Three-Dimensional Multiphase Flow Using a Level Contour Reconstruction Method for Front Tracking without Connectivity. *Journal of Computational Physics*, 180:427–470, 2002.
- [133] SHIN, S., ABDEL-KHALIK, S., DARU, V., AND JURIC, D. Accurate representation of surface tension using the level contour reconstruction method. *Journal of Computational Physics*, 203(2):493–516, 2005.
- [134] SIRIGNANO, W. A. Volume averaging for the analysis of turbulent spray flows. *International Journal of Multiphase Flow*, 31(6):675–705, 2005.
- [135] SMEREKA, P. *Level Set Methods for Two-Fluid Flows*, 1996.
- [136] SOVOVÁ, H. AND PROCHÁZKA, J. Breakage and coalescence of drops in a batch stirred vessel—I Comparison of continuous and discrete models. *Chemical Engineering Science*, 36(1):163–171, 1981.
- [137] SRINIVASAN, V., SALAZAR, A. J., AND SAITO, K. Numerical Investigation on the Disintegration of Round Turbulent Liquid Jets using LES/VOF Techniques. *Atomization and Sprays*, 18:571–617, 2008.

- [138] STONE, H. A. AND LEAL, L. G. Relaxation and breakup of an initially extended drop in an otherwise quiescent fluid. *Journal of Fluid Mechanics*, 198:399–427, 1989.
- [139] STONE, H. A., BENTLEY, B. J., AND LEAL, L. G. An experimental study of transient effects in the breakup of viscous drops. *Journal of Fluid Mechanics*, 173:131–158, 1986.
- [140] STRAIN, J. Tree Methods for Moving Interfaces. *Journal of Computational Physics*, 151(2):616–648, 1999.
- [141] SUSSMAN, M. A second order Coupled Level Set and Volume-of-Fluid Method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, 187:110–136, 2003.
- [142] SUSSMAN, M. AND FATEMI, E. An Efficient, Interface-Preserving Level Set Re-distancing Algorithm and Its Application to Interfacial Incompressible Fluid Flow. *SIAM Journal on Scientific Computing*, 20(4):1165–1191, 1999.
- [143] SUSSMAN, M. AND PUCKETT, E. A coupled level set and Volume-of-Fluid method for computing 3d and axisymmetric incompressible two-phase flows. *J. Comp. Phys.*, 162:301–337, 2000.
- [144] SUSSMAN, M., SMEREKA, P., AND OSHER, S. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. *Journal of Computational Physics*, 114(1):146–159, 1994.
- [145] SUSSMAN, M., FATEMI, E., SMEREKA, P., AND OSHER, S. An Improved Level Set Method for Incompressible Two-Phase Flows. *Computers & Fluids*, 27:663–680, 1998.
- [146] SUSSMAN, M., ALMGREN, A. S., BELL, J. B., COLELLA, P., HOWELL, L. H., AND WELCOME, M. L. An Adaptive Level Set Approach for Incompressible Two-Phase Flows. *Journal of Computational Physics*, 148(1):81–124, 1999.
- [147] SUSSMAN, M., SMITH, K., HUSSAINI, M., OHTA, M., AND ZHI-WEI, R. A sharp interface method for incompressible two-phase flows. *Journal of Computational Physics*, 221(2):469–505, 2007.
- [148] TAKAGI, S. AND MATSUMOTO, Y. Surfactant Effects on Bubble Motion and Bubbly Flows. *Annual Review of Fluid Mechanics*, 43(1):615–636, 2011.
- [149] TALEBANFARD, N. AND BOERSMA, B. J. Direct Numerical Simulation of Heat Transfer in Colliding Droplets by a Coupled Level Set and Volume of Fluid Method. In Fröhlich, J., Kuerten, H., Geurts, B. J., and Armenio, V., editors, *Direct and Large-Eddy Simulation IX*, number 20 in ERCOFTAC Series, pages 687–693. Springer International Publishing, 2015. DOI: 10.1007/978-3-319-14448-1_86.

BIBLIOGRAPHY

- [150] TANG, J. *Two-level preconditioned conjugate gradient methods with applications to bubbly flow problems*. PhD thesis, TU Delft, 2008.
- [151] TANGUY, S. AND BERLEMONT, A. Application of a level set method for simulation of droplet collisions. *International Journal of Multiphase Flow*, 31(9): 1015–1035, 2005.
- [152] TATEBE, O. The Multigrid Preconditioned Conjugate Gradient Method. In *Sixth Copper Mountain Conference on Multigrid Methods*, volume 3224, pages 621–634. NASA, 1993.
- [153] TRYGGVASON, G., BUNNER, B., ESMAEELI, A., JURIC, D., AL-RAWAHI, N., TAUBER, W., HAN, J., NAS, S., AND JAN, Y. J. A Front-Tracking Method for the Computations of Multiphase Flow. *Journal of Computational Physics*, 169(2):708–759, 2001.
- [154] TRYGGVASON, G., ESMAEELI, A., LU, J., AND BISWAS, S. Direct Numerical Simulations of gas/liquid multiphase flows. *Fluid Dynamics Research*, 38(9):660–681, 2006.
- [155] TRYGGVASON, G., THOMAS, S., LU, J., AND ABOULHASANZADEH, B. Multiscale issues in DNS of multiphase flows. *Acta Mathematica Scientia*, 30(2):551–562, 2010.
- [156] UHLMANN, M. Interface-resolved direct numerical simulation of vertical particulate channel flow in the turbulent regime. *Physics of Fluids*, 20(5):053305, 2008.
- [157] UNVERDI, S. O. AND TRYGGVASON, G. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100(1):25–37, 1992.
- [158] VAUDOR, G., BERLEMONT, A., MÉNARD, T., AND DORING, M. A Consistent Mass and Momentum Flux Computation Method Using Rudman-Type Technique With a CLSVOF Solver. In *ASME 14th International Symposium on Numerical Methods for Multiphase Flow*, page V01CT23A012, 2014.
- [159] VREEKEN, D. J. Direct Numerical Simulation of dilute droplet-laden flows. MSc Thesis, Delft University of Technology, 2011.
- [160] VUIK, C., SEGAL, A., AND MEIJERINK, J. A. An Efficient Preconditioned CG Method for the Solution of a Class of Layered Problems with Extreme Contrasts in the Coefficients. *Journal of Computational Physics*, 152(1):385–403, 1999.
- [161] WANG, L.-P. AND MAXEY, M. R. The motion of microbubbles in a forced isotropic and homogeneous turbulence. *Applied Scientific Research*, 51(1):291–296, 1993.
- [162] WANG, Z., YANG, J., AND STERN, F. Comparison of Particle Level Set and CLSVOF Methods for Interfacial Flows. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 2008.

- [163] WANG, Z., YANG, J., AND STERN, F. An improved particle correction procedure for the particle level set method. *Journal of Computational Physics*, 228(16):5819–5837, 2009.
- [164] WESSELING, P. *Principles of Computational Fluid Dynamics*. Springer, 1 edition, 2000.
- [165] YOUNGS, D. L. Time-Dependent Multi-Material Flow With Large Fluid Distortion. In *Numerical Methods for Fluid Dynamics Conference*, pages 273–285, 1982.
- [166] ZHANG, P. AND LAW, C. K. An analysis of head-on droplet collision with large deformation in gaseous medium. *Physics of Fluids*, 23(4):042102–22, 2011.
- [167] ZHOU, H. AND POZRIKIDIS, C. The flow of suspensions in channels: Single files of drops. *Physics of Fluids A: Fluid Dynamics*, 5(2):311, 1993.
- [168] ZHOU, H. AND POZRIKIDIS, C. The Flow of Ordered and Random Suspensions of Two-Dimensional Drops in a Channel. *Journal of Fluid Mechanics*, 255:103–127, 1993.



ACKNOWLEDGEMENTS

*The only people who you should get even with are those
who have helped you*

— John Southard

Het is goed dat men aan het begin van een onderzoekstraject niet weet waar het uitmondt en hoelang het zal duren. Hoewel ik zelf natuurlijk verantwoordelijk ben voor het uiteindelijke resultaat, had het onderzoek er heel anders uit gezien zonder de steun van een grote groep mensen.

Allereerst wil ik graag mijn promotor Bendiks Jan Boersma bedanken voor het mogelijk maken van een PhD positie waarbinnen ik mijn onderzoek kon doen. Mijn promotieonderzoek heb je op grote lijnen richting gegeven, terwijl je de dagelijkse begeleiding in handen liet bij mijn copromotor Wim-Paul Breugem. Deze opzet leverde een plezierige balans tussen mijn onderzoeksvrijheid en het inhoudelijk bijsturen waar nodig geacht. Jullie vakinhoudelijke kennis en analyse van problemen is bijzonder leerzaam geweest. Wim-Paul, bedankt voor je enthousiasme en betrokken begeleiding, het was plezierig dat je deur altijd voor mij openstond.

Voor problemen op IT-gebied kon ik altijd terecht bij Gerben Roest, waardoor deze vaak snel verholpen waren. Dankzij Willem Vermin van SARA kregen we inzicht in mogelijkheden voor efficiëntieverbeteringen, welke de basis vormen voor hoofdstuk 2. Binnen het lab was er de ondersteuning van Caroline en Ria op het secretariaat. Caroline, vooral bedankt voor je hulp tijdens het afronden van mijn thesis toen ik zelf niet meer in Delft aanwezig was.

Verder wil ik mijn kamergenoten Arnoud, Astrid en Daniël bedanken voor de gezellige gesprekken over een breed scala aan onderwerpen: van de perikelen rond de lichte mannen vier-zonder tot het wel en wee van kippenembryos. Als de onderwerpen echt moeilijk werden was er gelukkig altijd onze bijna-kamergenoot Ellert, die voor een kop koffie vanuit het Flowmotion HQ naar beneden kwam.

Leuke momenten van de dag waren ook de dagelijkse koffie en lunch pauzes. De sfeer hierbij was altijd goed, mede dankzij Aris, Carole, Christian Poelma, Christian Potma, Daniel, Daniele, Dirk Jan (bedankt dat je me in contact hebt gebracht met Bendiks Jan toen ik op zoek was naar een promotieplaats), Edwin, Gerrit, Gijs, Gosse, Hyungsoo, Jasper R, Jasper T, Jeanette, Jerke, Jerry, Joop, Manu, Marc, Marieke, Mark F, Mark T, Massi, Mathieu, Nikolay, Norbert, Pepijn, René, Rob, Ruud, Sasha, Sedat, Simon, Sita, Tjerk (het was een plezier je MSc onderzoek te begeleiden), Valentina, Uli en Vincent.

Na alle mentale arbeid in Delft had ik bij AV Phoenix in Utrecht een plezierige groep waar ik tijdens het hardlopen m'n hoofd even kon leegmaken. Hierbij bedank ik in het bijzonder mijn trainers Herman en Henny en de CWT-commissieleden (Arnoud, Frits, Patrick, Roelof en Simon) waarmee ik samen enkele edities van de Maliebaanloop heb georganiseerd. Via Phoenix heb ik ook Steven V leren kennen,

waarmee ik later maanden door Azië heb gereisd. En natuurlijk Maarten, die Steven en mij heeft vergezeld tijdens de prachtige Annapurna Circuit hike in Nepal en in het Spaanse Navàs. Beiden bedankt voor jullie interesse tijdens de laatste loodjes van mijn thesis. Steven, bedankt dat jij één van mijn paranimfen wil zijn.

Vlak voor aanvang van mijn promotieonderzoek heb ik samen met Laura een huis gekocht in Utrecht. Ieder beschikbaar uurtje hebben we eigenhandig het huis verbouwd tot wat het nu is. Laura, ik hoop dat je er nog lang met veel plezier zal wonen. Bedankt voor je steun en vertrouwen tijdens de eerste fase van mijn onderzoek, ik heb goede herinneringen aan onze (bouw)avonturen. Deze dank ben ik ook aan je familie verschuldigd, die altijd behulpzaam en geïnteresseerd zijn geweest in het reilen en zeilen van mijn onderzoek.

Dirk, Frank en Hans: al zien we elkaar niet meer zo regelmatig als vroeger, de nachtelijke uitstapjes naar de Stunt en d'Aesculaap zal ik niet vergeten. De jaarlijkse Adventure Run op Ameland heb ik wegens het reizen de afgelopen jaren aan me voorbij moeten laten gaan, maar dit jaar hoop ik weer van de partij te zijn! Frank, bedankt voor de bijzondere (road)trip door de VS, Mexico, Belize en Guatemala.

Auke & Ilona, Emile & Marit, Erik & Moniek, Frank & Eline, Nico & Maaïke en Steven: ik ben blij dat we, ondanks de verspreidheid over de wereld, elkaar als oud-medestudenten van Lucht- en Ruimtevaarttechniek nog regelmatig treffen. Ik heb goede herinneringen aan onze gezamenlijke studententijd, maar ook de jaren erna met trouwerijen en gezinsuitbreidingen. Ik weet niet of er op mijn promotie nog flessen drank zijn verweerd, maar ik drink er graag één met jullie op. 'Broer' Emile, bedankt dat jij vandaag als één van mijn paranimfen wilt optreden.

Natuurlijk mag in dit dankwoord ook mijn familie niet ontbreken. Esther & Hans, Jeroen & Mirjam, Petra & Floris bedankt voor jullie interesse in mijn onderzoek. Ook mijn ouders, Theo & Coby, hebben me op vele manieren gesteund. Van jongs af aan hebben ze me altijd weten te stimuleren om doelen te halen, en zonder hen was ik dan ook zeker niet zover gekomen. Zelfs op momenten dat ik afweek van de gangbare paden en het roer omgooide kreeg ik jullie steun. Dank daarvoor, dit maakte het hele proces een stuk makkelijker. Het was weer een onvergetelijk familieweekend met z'n allen in Bad Bentheim, hopelijk volgen er nog vele.

Finalmente, mi querida Paula! De un festival de la cerveza en Cabo San Lucas a una promoción en Delft, puede sonar raro. A pesar de la distancia en el último año y medio hemos pasado mucho tiempo especial juntos. Tengo una sensación de felicidad cuando pienso en todas esas hermosas experiencias y espero que vengan muchas más. Gracias por tus palabras de aliento y tus entrenamientos de Pilates en los momentos en que necesitaba distraerme un poco. Tú y tu familia son como un segundo hogar (país), estoy contento y deseando visitarlos en México muy pronto.

Waarschijnlijk ben ik nog wel wat mensen persoonlijk vergeten te noemen: onder andere ooms en tantes, buren van Klein Doesburg, en oud-collega's van KWR. Iedereen die het gevoel heeft ook thuis te horen in dit dankwoord: bedankt!

Marcel Kwakkel
Kopenhagen, mei 2016

ABOUT THE AUTHOR

Education is an admirable thing, but it is well to remember from time to time that nothing that is worth knowing can be taught

— Oscar Wilde

Marcel Kwakkel was born on June 24, 1981 in Heerde, The Netherlands. After finishing secondary school at CC De Noordgouw in 1999, he started studying Aerospace Engineering at Delft University of Technology with a Master specialization in Computational Fluid Dynamics. In 2004 he spent six months at Daimler-Chrysler in Sindelfingen, Germany for his internship entitled '*Flow investigation on the 1:4 scale W203 model in varying crosswind conditions – Influence of a spoiler and cooling air flow on the stability*'. Next, he started working on his Master project at the Aerodynamics department of the faculty of Aerospace Engineering under supervision of Dr. ir. M.I. Gerritsma. The topic of this research was '*Time dependent flow simulations using the least-squares spectral element method – Application to unsteady incompressible Navier-Stokes flows*', which resulted in his MSc degree in June 2007.

Since Marcel was interested in pursuing a PhD in Fluid Dynamics, he filled a temporary position at Delft University of Technology until an interesting vacancy became available. In July 2007 he bought a house in Utrecht which required a lot of reconstruction, on which he spend most of his spare time until 2012. In June 2008 he started his PhD research under supervision of Dr. W.P. Breugem and Prof. B.J. Boersma at the Laboratory of Aero- and Hydrodynamics at Delft University of Technology. During the summer of 2010 he attended the *CSCAMM Tutorial School on Fluid Dynamics: Topics in Turbulence* at the Univeristy of Maryland and subsequently visited Prof. Sussman at the Carnegie Mellon University in Pittsburgh for three weeks. The work carried out in this PhD project has finally led to this thesis '*Numerical modeling of turbulent dense dispersed two-phase flows*'.

After ending his contract at Delft University of Technology in September 2012, Marcel started a job as Scientific Researcher at KWR Watercycle Research Institute in Nieuwegein. After almost a year of successfully working in that position, he decided to sell all his material belongings and travel the world. From that moment on he backpacked through Asia (October 2013-May 2014), Europe (June-August 2014), North and Central America (August 2014-May 2015), Europe (June-November 2015), Mexico (November 2015-January 2016), Morocco (January-February 2016), Spain (February-March 2016), and some shorter trips since then. During his travels he got to know many different cultures, learned to solve a Rubik's cube within 90 seconds, crossed the Himalayas, became a certified scuba diver, studied Spanish, improved his programming skills and finished this thesis.

