# Computation of Bubbly Flows with a Mass-Conserving Level-Set Method

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.dr.ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 22 november 2005 om 10:30 uur

door

Sander Pieter VAN DER PIJL
ingenieur luchtvaart en ruimtevaart

geboren te Ter Aar.

Dit proefschrift is goedgekeurd door de promotor:
Prof.dr.ir. P. Wesseling

Samenstelling promotiecomissie:

| | |
|---|---|
| Rector Magnificus, | voorzitter |
| Prof.dr.ir. P. Wesseling, | Technische Universiteit Delft, promotor |
| Dr.ir. C. Vuik, | Technische Universiteit Delft, toegevoegd promotor |
| Prof.dr.ir. H.W.M. Hoeijmakers, | Universiteit Twente |
| Prof.dr.ir. B. Koren, | Technische Universiteit Delft |
| Prof.dr.ir. J.A.M. Kuipers, | Universiteit Twente |
| Prof.dr.ir. A.E.P. Veldman, | Rijksuniversiteit Groningen |
| Dr.ir. B.J. Boersma, | Technische Universiteit Delft |

Computation of Bubbly Flows with a Mass-Conserving Level-Set Method.
Dissertation at Delft University of Technology.
Copyright © 2005 by S.P. van der Pijl.

NWO
Netherlands Organisation for Scientific Research

The work described in this thesis was financially supported by the Netherlands Organization for Scientific Research (NWO).

ISBN 90-8559-119-8

# Summary

The subject of this thesis is the computation of bubbly flows. Bubbly flows occur for example in chemical reactors, boiling, fuel injectors and coating. The bubbles and the surrounding fluid are modeled directly without phase averaging. This is the most fundamental approach to the mathematical modeling of two-phase flows. Every bubble is modeled in great detail. Since the model is so demanding, an efficient numerical approach has to be taken. This work aims to develop an efficient, robust method for the direct numerical simulation of two-phase flows.

We consider two incompressible fluids (e.g. air and water) that are separated by an interface. The interface is a moving, internal boundary, where density and viscosity change discontinuously and surface tension forces act. Since the interface moves with the local velocity, the location of the interface is part of the problem. This makes it a moving (internal) boundary problem. The fluids on either side of the interface are, mathematically, coupled by interface conditions.

When the governing equations are discretized, the question arises how to deal with this interface. Various methods have been put forward to treat moving boundary problems. The two methods that are of most interest are the so-called Volume-of-Fluid method and the Level-Set method. With both methods, a coloring function is used to identify the individual phase. The interface is implicitly defined, which allows arbitrarily shaped interfaces and topology changes (break-up and merging). The interface is evolved by advecting the coloring function in the flow field as if it is a material property. Due to the specific choice of the coloring function, the Level-Set method has some advantages over the Volume-of-Fluid method and has been chosen as the basis of this work. However, the Level-Set method does not conserve the mass of the individual phase, which is considered the major drawback of the method. The Volume-of-Fluid method on the other hand is rigorously mass conserving, but difficult to implement. Therefore the coloring function of the Volume-of-Fluid method (the Volume-of-Fluid *function*) is only used to conserve mass. This is achieved without encountering the difficulties associated with the Volume-of-Fluid method.

The strategy of modeling bubbly flows is to compute the flow with a given interface position and to subsequently evolve the interface in the given flow field. The governing equations are discretized by a finite difference approach on a Cartesian grid in a rectangular domain. The viscosity is regularized to decouple the interface conditions. For the interface forces the Continuous Surface Force approach is adopted, where the interface forces are smeared out near the interface. The Ghost-Fluid method for incompressible flows is used to compute derivatives near the interface. The pressure correction method is used for the temporal discretization of the flow equations.

The advection of the interface is the key part of this thesis, for which the Level-Set approach is chosen. To ensure mass conservation, additional effort is necessary and in this research the Volume-of-Fluid function is used to conserve mass. Firstly, pure Level-Set advection is applied. Then, in order to conserve mass, corrections to the Level-Set function are made by using a Volume-of-Fluid

function. For that purpose an explicit relation between the Level-Set function and the Volume-of-Fluid function is derived. This relation is also used to advect the Volume-of-Fluid function, which circumvents the difficult convection step of the Volume-of-Fluid method. A simple iterative procedure is used to find the corrections to the Level-Set function, such that mass is conserved. Operator splitting is employed in the advection step and can, unfortunately, cause unphysical values of the Volume-of Fluid function. This is addressed by means of mass redistribution, where erroneous mass is transported towards the nearest interface location.

Since interface forces and viscosity are regularized near the interface, the Level-Set function is made a distance function every time-step by performing a re-initialization procedure. The difficulty with re-initialization is that the interface shifts during the process and hence (additional) mass errors are made. The re-initialization procedure is modified to overcome this.

The curvature of the interface has to be computed for the modeling of surface tension effects. This means that first and second order derivatives of the Level-Set function need to be determined. Since local corrections to the Level-Set function are made near the interface in order to conserve mass, problems arise when these derivatives are computed. This results in parasitic currents. To reduce these currents, the corrections to the Level-Set function are made smaller by introducing a start criterion in the re-initialization procedure. Besides that, the currents are reduced even further by smoothing of the computed curvature.

The behavior of the resulting mass-conserving Level-Set approach is investigated by computing several standard advection test problems with a prescribed velocity field. Results are compared with pure, high order Level-Set advection. The method is applied to the complete set of equations by considering a falling drop and a rising bubble in two and three dimensions, respectively. Merging of rising bubbles is studied for two aligned and two misaligned bubbles. Comparison with other numerical work and experimental data is made.

This thesis finishes with concluding remarks and recommendations for future research.

# Berekeningen van Stromingen met Bellen met een Massa-Behoudende Level-Set Methode

## Samenvatting

Het onderwerp van dit proefschrift is de berekening van stromingen met bellen. Stromingen met bellen komen voor in bijvoorbeeld chemische reactors, koken, brandstofinspuiting en coating. De bellen en omringende vloeistof worden direct gemodeleerd zonder fase-middeling. Dit is de meest fundamentele aanpak om tweefase stromingen wiskundig te beschrijven. Elke bel wordt in detail gemodeleerd. Omdat het model zo veeleisend is moet een efficiënte numerieke aanpak genomen worden. Dit werk richt zich op de ontwikkeling van een efficiënte, robuste methode voor de directe numerieke simulatie van tweefase stromingen.

We beschouwen twee incompressibele vloeistoffen (bijv. lucht en water) die gescheiden worden door een interface. Het interface is een bewegende, inwendige rand, waar dichtheid en viscositeit discontinu veranderen en oppervlaktespannings krachten werken. Omdat het interface met de lokale snelheid meebeweegt, is de positie van het interface een deel van het probleem. Dit maakt het tot een bewegend (inwendige) rand probleem. De vloeistoffen aan elke kant van het interface zijn, wiskundig gezien, gekoppeld door interface condities.

Wanneer de beschrijvende vergelijkingen gediscretiseerd worden, rijst de vraag hoe met het interface omgaan moet worden. Er zijn verschillende methoden voor bewegende-rand problemen. De twee interessantste methoden zijn de zogenaamde Volume-of-Fluid en Level-Set methode. Bij beide methoden wordt een markeringsfunctie gebruikt om elke afzonderlijke fase aan te geven. Het interface is impliciet gedefinieerd, wat willekeurig gevormde interfaces en veranderingen van topologie (uiteen breken en fuseren) toestaat. Het interface wordt verplaats door de markeringsfunctie te advecteren in het stromingsveld alsof het een materiaaleigenschap is. Door de specifieke keuze van de markeringsfunctie heeft de Level-Set methode voordelen ten opzichte van de Volume-of-Fluid methode. Daarom is de Level-Set methode gekozen als basis voor dit werk. Echter, de Level-Set methode behoudt niet de massa van elke afzonderlijke fase; dit wordt gezien als het voornaamste nadeel van deze methode. Aan de andere kant is de Volume-of-Fluid methode strikt massa-behoudend, maar moeilijk om te implementeren. Daarom is de markeringsfunctie van de Volume-of-Fluid methode (de Volume-of-Fluid *functie*) alleen gebruikt om massa te behouden. Dit is bewerkstelligd onder vermijding van de moeilijkheden die verbonden zijn aan de Volume-of-Fluid methode.

De strategie voor het modeleren van stromingen met bellen is om de stroming te berekenen met een gegeven interface positie en vervolgens het interface te verplaatsten in het gegeven stromingsveld. De beschrijvende vergelijkingen worden gediscretiseerd met eindige differenties op een Cartesisch rooster in een rechthoekig domein. De viscositeit is geregulariseerd om de interface condities te ontkoppelen. Voor de interface krachten is de 'Continuous Surface Force' aan-

pak gebruikt, waarbij de interface krachten uitgesmeerd worden bij het interface. De ‘Ghost-Fluid’ methode voor incompressibele stromingen wordt gebruikt om afgeleiden van de druk bij het interface uit te rekenen. De ‘pressure correction’ methode wordt gebruikt bij de tijdsdiscretisatie van de stromingsvergelijkingen.

Het belangrijkste deel van deze dissertatie is de advectie van het interface, waarvoor de Level-Set aanpak gekozen is. Er is extra inspanning nodig om massa te behouden en in dit werk wordt daarvoor de Volume-of-Fluid functie gebruikt. Eerst wordt een pure Level-Set advectie stap gedaan. Daarna worden om massa te behouden correcties gemaakt op de Level-Set functie door gebruik te maken van de Volume-of-Fluid function. Hiervoor wordt een expliciete relatie afgeleid tussen de Level-Set functie en de Volume-of-Fluid functie. Deze relatie wordt tevens gebruikt om de Volume-of-Fluid functie te advecteren, waarmee de moeilijke advectie stap van de Volume-of-Fluid methode vermeden wordt. Een simpele iteratie procedure wordt gebruikt om correcties op de Level-Set functie te vinden, zodanig dat massa behouden blijft. In de advectie stap wordt de operator gesplitst en dit kan, helaas, ongeldige waarden van de Volume-of-Fluid veroorzaken. Dit wordt verholpen door massa redistributie, waarbij foutieve massa getransporteerd wordt in de richting van het meest nabije interface.

Omdat de interface krachten en viscositeit geregulariseerd worden bij het interface, wordt van de Level-Set functie voor elke tijdstap een afstandsfunctie gemaakt door een reïninitialisatie procedure uit te voeren. De moeilijkheid met reïnitialisatie is dat het interface tijdens het proces verschuift, waardoor (extra) massa verloren of gewonnen wordt. Om dit te verhelpen is de reïnitialisatie procedure aangepast in Hoofdstuk 6.

De kromtestraal van het interface moet uitgerekend worden om effecten van oppervlaktespanning te kunnen modeleren. Dit betekent dat eerste en tweede afgeleiden van de Level-Set functie bepaald moeten worden. Omdat lokaal bij het interface correcties op de Level-Set functies gemaakt worden om massa te behouden, rijzen er problemen wanneer die afgeleiden berekend worden. Dit resulteert in parasitaire stromingen. Om die stromingen te reduceren worden de correcties op de Level-Set functie kleiner gemaakt door het introduceren van een begin criterium in de reïnitialisatie procedure. Bovendien worden de parasitaire stromingen nog verder gereduceerd door het gladstrijken van de berekende kromtestralen.

Het gedrag van de massa-behoudende Level-Set aanpak wordt geïllustreerd aan de hand van standaard advectie tests met een voorgeschreven snelheidsveld. De resultaten worden vergeleken met pure, hogere orde Level-Set advectie. De methode wordt toegepast op de volledige set vergelijkingen door te kijken naar een vallende druppel en stijgende bel in twee en drie dimensies, respectievelijk. Het samengaan van stijgende bellen wordt bestudeerd met twee uitgelijnde bellen en twee niet-uitgelijnde bellen. Er wordt vergeleken met ander numeriek werk en experimentele data.

Deze dissertatie eindigt met afsluitende opmerkingen en aanbevelingen voor toekomstig onderzoek.

# Contents

**Summary** **iii**

**Samenvatting** **v**

**List of Figures** **xi**

**List of Tables** **xv**

**1 Introduction** **1**
1.1 Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

**2 Mathematical Model** **5**
2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
2.2 Governing equations . . . . . . . . . . . . . . . . . . . . . . . . . . 5
2.3 Surface tension . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
2.4 Interface boundary conditions . . . . . . . . . . . . . . . . . . . . . 7
2.5 Regularization of viscosity . . . . . . . . . . . . . . . . . . . . . . . 8
2.6 Continuous Surface Force (CSF) . . . . . . . . . . . . . . . . . . . 8
2.7 Initial and Boundary conditions . . . . . . . . . . . . . . . . . . . . 10

**3 Moving Boundary methods** **11**
3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
3.2 Capturing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
3.3 Front tracking . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
3.4 Volume tracking . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
  3.4.1 Volume-of-Fluid method . . . . . . . . . . . . . . . . . . . . 14
  3.4.2 Level-Set method . . . . . . . . . . . . . . . . . . . . . . . . 17
3.5 Combined methods . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
3.6 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

**4 Numerical Approach** **23**
4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
4.2 Regularization of viscosity . . . . . . . . . . . . . . . . . . . . . . . 23
4.3 Continuous Surface Force . . . . . . . . . . . . . . . . . . . . . . . 24
4.4 Spatial discretization . . . . . . . . . . . . . . . . . . . . . . . . . . 25

4.5 Temporal discretization . . . 27
4.6 Initial and Boundary conditions . . . 30
4.7 Time Step criteria . . . 31
4.8 Conclusion . . . 31

**5 Interface advection 33**
5.1 Introduction . . . 33
5.2 Level-Set advection . . . 33
5.3 Re-initialization . . . 34
5.4 Mass-Conserving Level-Set advection . . . 36
  5.4.1 Step 1: Volume-of-Fluid function . . . 37
  5.4.2 Step 2: Volume-of-Fluid advection . . . 44
  5.4.3 Step 3: Inverse function . . . 46
  5.4.4 Mass redistribution . . . 47
5.5 Conclusion . . . 51

**6 Improved re-initialization 53**
6.1 Introduction . . . 53
6.2 Modified re-initialization . . . 54
6.3 Stop criterion . . . 56
6.4 Conclusion . . . 57

**7 Surface tension 59**
7.1 Introduction . . . 59
7.2 Discretization of curvature . . . 59
7.3 Parasitic currents . . . 60
7.4 Re-initialization start criterion . . . 61
7.5 Effect of mass redistribution . . . 61
7.6 Curvature smoothing . . . 66
7.7 Conclusion . . . 69

**8 Applications 71**
8.1 Introduction . . . 71
8.2 Two-dimensional results . . . 71
  8.2.1 Linear advection . . . 71
  8.2.2 Zalesak's rotating disc . . . 72
  8.2.3 Air/water flow . . . 77
  8.2.4 Computational Costs . . . 78
8.3 Three-dimensional results . . . 81
  8.3.1 Advection test . . . 81
  8.3.2 Rising air bubble in water without surface tension . . . 81
  8.3.3 Rising air bubble in water with surface tension . . . 90
  8.3.4 Falling water droplet in air without surface tension . . . 92
  8.3.5 Falling water droplet in air with surface tension . . . 93
  8.3.6 Comparison of rising bubble with experimental data . . . 100
  8.3.7 Coalescence of two aligned rising bubbles . . . 103

8.3.8 Coalescence of two misaligned rising bubbles . . . . . . . 106

**9 Conclusions** **113**

9.1 Recommendations for future research . . . . . . . . . . . . . . . . . 114

**Curriculum Vitae** **121**

**List of Publications** **123**

# List of Figures

1.1 Coalescence of two bubbles . . . . . . . . . . . . . . . . . . . . . . 2

2.1 Color function $\chi$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
2.2 Surface tension acting on interface element $A$ . . . . . . . . . . . 7
2.3 Regularized color function $\chi_\alpha$ . . . . . . . . . . . . . . . . . . . . . . 9
2.4 Regularized Dirac delta function $\delta_\alpha$ . . . . . . . . . . . . . . . . . 10

3.1 Marker particles; Marker-and-Cell method . . . . . . . . . . . . . 14
3.2 One-dimensional example of fractional volume $\Psi$; loss of interface definition after time step . . . . . . . . . . . . . . . . . . . . . . . 15
3.3 Donating region (hatched) defined by instantaneous streamlines (dashed) for right-hand-side face of a mixed cell . . . . . . . . . . 16
3.4 Interface reconstruction; VOF/PLIC . . . . . . . . . . . . . . . . 17
3.5 One-dimensional example of the level-set function $\Phi$; mass errors after time step . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
3.6 Adaptive grid refinement . . . . . . . . . . . . . . . . . . . . . . . . 19

4.1 Grid layout in two dimensions . . . . . . . . . . . . . . . . . . . . . 25
4.2 Ghost Fluid method; real values (gray) and ghost values (white) 25
4.3 Grid layout near the boundary in two dimensions . . . . . . . . . 30

5.1 Topologies of the cut plane for $\Phi \leq 0$ . . . . . . . . . . . . . . . . 39
5.2 Volumes for $\Phi \leq 0$ . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
5.3 Examples of Volume-of-Fluid function for the two cases . . . . . 41
5.4 Examples of the Volume-of-Fluid function . . . . . . . . . . . . . 43
5.5 Donating regions for fluxes $F_x$ and $F_y$. These are the shaded areas that will flow through the boundaries during a time step. Doubly fluxed areas exist near the corners of the cell. . . . . . . . 45
5.6 MCLS method: interface advection; $\Phi$: Level-Set function; $\Psi$: Volume-of-Fluid function. The left-hand side branch corresponds to pure Level-Set advection. The right-hand side branch represents the Volume-of-Fluid advection. . . . . . . . . . . . . . . . . . . . . . . . 48

6.1 Prescribed interfaces on a 20 × 20 mesh . . . . . . . . . . . . . . 54
6.2 Effect of re-initialization on small features . . . . . . . . . . . . . 55

6.3 Effect of modified re-initialization on small entities . . . . . . . . 56

7.1 Local extremum of the Level-Set function $\Phi$ between two approaching interfaces . . . . . . . . . . . . . . . . . . . . . . . . . 60
7.2 Parasitic currents for the Laplace test-case with three different grids . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
7.3 Parasitic currents for the Laplace test-case in the symmetry plane; $96^3$ grid . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
7.4 Parasitic currents for the Laplace test-case with three different grids; start criterion in re-initialization . . . . . . . . . . . . . . . 64
7.5 One-dimensional example of a Level-Set function $\Phi$ that does not change sign in $\Omega_k$ (hence no interface); but the linearization of $\Phi$ around $\mathbf{x}_k$ does, hence $0 < \Psi_k < 1$ and the classification 'numerical vapor'. . . . . . . . . . . . . . . . . . . . . . . . . . 65
7.6 Parasitic currents for the Laplace test-case with three different grids; start criterion in re-initialization and no mass redistribution 67
7.7 Parasitic currents for the Laplace test-case with three different grids; time-step criterion in re-initialization and smoothed curvature . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 68

8.1 Linear advection test. A disc of radius $R_0$ is advected in a rectangular domain of $L_x$ by $L_y$. The center of the disc is initially placed at $2R_0$ from the top of the domain. The advection velocity is $u$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 72
8.2 Relative mass errors for the linear advection test; $\epsilon = 10^{-8}$ (every $10^{th}$ iteration marked). Pure Level-Set advection with different discretization orders are compared with MCLS. . . . . . . . . . . 73
8.3 Zalesak's slotted disc advection test (to scale). The dimensions of the slot are depicted in the graph. . . . . . . . . . . . . . . . . 74
8.4 Results for Zalesak's advection test; The shaded area indicates the initial contour. The dashed lines indicate the interface after one revolution with $3^{rd}$ order pure Level-Set advection. The solid lines correspond to MCLS advection. Four different mesh sizes have been employed. . . . . . . . . . . . . . . . . . . . . . . . . . 75
8.5 Relative masses for Zalesak's advection test; $\epsilon = 10^{-8}$ (every $50^{th}$ iteration marked). Pure Level-Set advection with third order discretization is compared with MCLS for different mesh sizes. . 76
8.6 Errors for Zalesak's advection test. The left graph show the errors of the Level-Set-function $\Phi$ near the interface after one revolution. The right graph shows the errors in the computed interface length. 78
8.7 Interface positions for the rising bubble. Three different mesh sizes have been employed: $-\cdot-$ : $30\times45$; $--$ : $40\times60$; — : $60\times90$ mesh. Snapshots are presented with equidistant time-steps. . . . 79
8.8 Interface positions for the falling droplet; Three different mesh sizes have been employed: $-\cdot-$ : $30\times45$; $--$ : $40\times60$; — : $60\times90$ mesh. Snapshots are presented with equidistant time-steps. . . . 80

8.9 Relative mass errors for advected sphere; $\epsilon = 10^{-8}$ . . . . . . . . 82
8.10 Set-up of the rising-bubble test case . . . . . . . . . . . . . . . . 83
8.11 Rising bubble without surface tension; $64^3$ grid . . . . . . . . . . 84
8.12 Rising bubble without surface tension; $96^3$ grid . . . . . . . . . . 85
8.13 Rising bubble with low surface tension; courtesy of Sussman et al. [1] . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
8.14 Rising bubble without surface tension; $96^3$ grid; zoomed in . . . 87
8.15 Rise speed of the bubble without surface tension; the dotted line is the analytic rise speed of the spherical cap . . . . . . . . . . . 88
8.16 Bubble without surface tension just before it breaks through the free surface; symmetry plane $y = \frac{1}{2}L_y$; $R_s$ is the radius of the spherical cap. . . . . . . . . . . . . . . . . . . . . . . . . . . 89
8.17 Rising bubble with surface tension; $64^3$ grid . . . . . . . . . . . . 90
8.18 Rising bubble with surface tension; $96^3$ grid . . . . . . . . . . . . 91
8.19 Rise speed of the bubble with surface tension; marks are at equally spaced time intervals of 0.005 $s$. . . . . . . . . . . . . . . 92
8.20 Set-up of the falling-droplet without surface tension test case . . 93
8.21 Falling droplet without surface tension; $64^3$ grid . . . . . . . . . . 94
8.22 Falling droplet without surface tension; $96^3$ grid . . . . . . . . . . 95
8.23 Fall speed of the droplet without surface tension compared with the free-fall velocity . . . . . . . . . . . . . . . . . . . . . . . . 96
8.24 Falling droplet with surface tension; $64^3$ grid . . . . . . . . . . . 97
8.25 Falling droplet with surface tension; $96^3$ grid . . . . . . . . . . . 98
8.26 Fall speed of the droplet with surface tension compared with the free-fall velocity . . . . . . . . . . . . . . . . . . . . . . . . . . 99
8.27 Set-up of the rising bubble test case . . . . . . . . . . . . . . . . 101
8.28 Rise speed of the bubble; Eö = 1.0; $\log(M) = -3.8$ . . . . . . . . 102
8.29 Set-up of the two aligned rising-bubbles test case . . . . . . . . . 104
8.30 Coalescence of two aligned rising bubbles; $96 \times 96 \times 192$ grid . . 105
8.31 Coalescence of two aligned rising bubbles; courtesy of De Sousa et al. [2] . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 106
8.32 Rise speed of the aligned bubbles; $96 \times 96 \times 192$ grid . . . . . . . 107
8.33 Set-up of the two misaligned rising bubbles . . . . . . . . . . . . 108
8.34 Coalescence of two misaligned rising bubbles; $96 \times 96 \times 192$ grid 109
8.35 Coalescence of two misaligned rising bubbles; courtesy of De Sousa et al. [2] . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 110
8.36 Rise speed of the bubbles; $96 \times 96 \times 192$ grid . . . . . . . . . . . 111
8.37 Location of the center of gravity $(x_c, z_c)$ of the rising bubbles in the $x - z$ symmetry plane; marks are at equally spaced time intervals of 0.02 $s$; $96 \times 96 \times 192$ grid . . . . . . . . . . . . . . . . 112

# List of Tables

3.1 Classification of free-boundary methods . . . . . . . . . . . . . . . 12
3.2 Methodology of PLIC . . . . . . . . . . . . . . . . . . . . . . . . . 17

8.1 Computed interface lengths after one revolution . . . . . . . . . . . 77
8.2 Computational costs measured in CPU seconds per time-step. MCLS is compared with third order Level-Set advection for the rising bubble test case on a 60 × 90 mesh. . . . . . . . . . . . . . . 81
8.3 Reynolds numbers of the rising bubble compared with the numerical results from Gunsing [3] and experimental data; Eö = 1.0; $\log(M) = -3.8$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . 103

# Chapter 1

# Introduction

Modeling of multi-phase flows is a major challenge for Computational Fluid Dynamics (CFD). Multi-phase flows occur commonly in engineering fluid mechanics. Chemical reactors, boiling, fuel injectors and coating are a few examples from petrochemical industry. Naval engineering is another example, where complex multi-phase flows have to be addressed, mostly in relation to the phenomenon of cavitation.

It will be clear from the preceding examples that multi-phase flows cover a wide range of application areas. Although mathematically equivalent, there exist large differences in characteristic length, time and velocity scales. From a modeling point of view it is important to firstly narrow down the region of interest: in this thesis the modeling of small-scale two-phase flows, such as falling water droplets in air and rising air bubbles in water, are considered. This is relevant for e.g. bubble columns, that are often used as a mixing and mass transfer device in chemical industry. To improve their design, a sound knowledge of the dynamics of individual bubbles is necessary.

There exist a hierarchy of multi-phase flow models. The highest level is the multi-fluid or two-fluid model. A continuum approach is taken based on the concentrations of the phases within a control volume. The governing equations are obtained by volume averaging. Small-scale information is lost and additional modeling or 'constitutive equations' are needed for the closure of the model. The constitutive equations are uncertain and form the weak point of the two-fluid model. In principle an arbitrary number of bubbles or droplets can be modeled, as long as the continuity hypothesis holds. This makes the two-fluid model interesting for industrial applications.

A lower level approach is the 'Discrete Bubble Model' where a continuum approach is used for one phase (e.g. water) and particles are used to model the bubbles. Constitutive relations are needed that express the forces acting on the bubbles. These forces are complex and the formulation is not well-established. Since every individual droplet or bubble is tracked, the maximum number is limited.

The lowest level approach is the direct description of the individual phases.

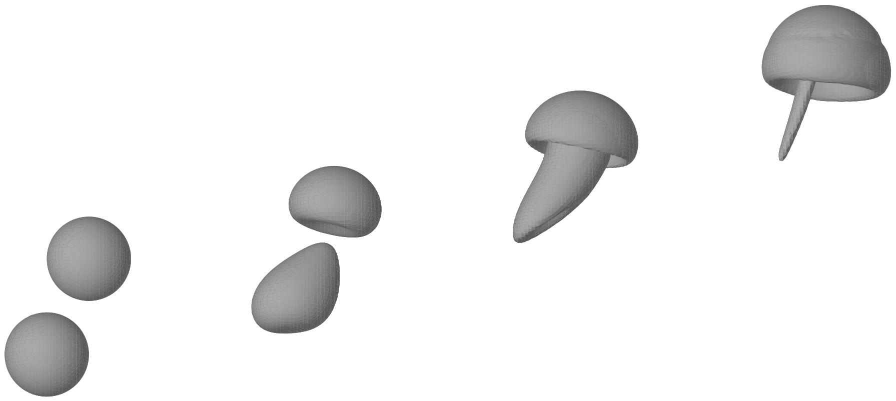


Figure 1.1: Coalescence of two bubbles

No closure modeling is needed. Only a small number of bubbles can be modeled, since each bubble is modeled in great detail. The direct description is therefore not commonly used for industrial applications. It is used for the validation of the closure relations of the higher-level models. On the other hand, such small scales do not require any kind of turbulence modeling as the flow is laminar. Since the model is so demanding, an efficient numerical approach has to be taken. This work aims to develop an efficient, robust method for the direct numerical simulation (DNS) of multi-phase flows.

The direct modeling of the two phases is as follows: there exist two incompressible fluids (e.g. air and water) that are separated by an interface. The interface is a moving, internal boundary, where density and viscosity change discontinuously and surface tension forces act. Mathematically speaking, this is expressed by interface conditions.

The dynamics of the flow are governed by buoyancy and surface tension effects. The bubbles might collide and brake up. This means that the topology of the interface is arbitrary. An example is shown in Fig. 1.1, where rising bubbles collide. A numerical method has to be able to locate the interface, advect it and apply the interface conditions. The focus of this research is on a efficient, robust way to deal with the interface.

## 1.1 Outline

The mathematical model is described in Chapter 2. The governing equations and interface conditions will be presented. The treatment of the interface discontinuities and surface tension forces is given and motivated.

In Chapter 3 a review of moving boundary methods that are available from literature is presented. This will mainly focus on the ‘Volume-of-Fluid’ method and the ‘Level-Set’ method. These two methods are of most interest for this research. A motivation for the Level-Set method as basis of this work will be given. It will become clear that the major disadvantage of the Level-Set method is that it does not conserve the mass of each individual phase.

The numerical approach is presented in Chapter 4. It deals with the discretization of the flow equations, application of the interface conditions and time integration. The interface advection is discussed in Chapter 5. Firstly the pure Level-Set advection is studied. Thereafter the ‘Mass Conserving Level-Set’ method is presented.

So-called re-initialization is commonly used in Level-Set methods. A simple modification of the original procedure, such that it suits the ‘Mass Conserving Level-Set’ method better, is discussed in Chapter 6. Incorporation of surface tension is the subject of Chapter 7. The effect of the ‘Mass-Conserving Level-Set’ method is investigated by considering the *Laplace test-case*.

Applications are shown in Chapter 8. Two- and three-dimensional test cases are included. Advection tests are used to show the properties of the ‘Mass-Conserving Level-Set’ approach. The method is applied to a falling droplet and rising bubble in two and three dimensions. Merging of bubbles is illustrated by considering two aligned and two misaligned rising bubbles.

# Chapter 2

# Mathematical Model

## 2.1 Introduction

In this research the modeling of two incompressible fluids that are separated by an interface is considered. The flow is described by a velocity field **u** and pressure $p$. Since the interface moves with the local velocity **u**, the location of the interface is part of the problem. This makes it a moving (internal) boundary problem.

The fluids on either side of the interface are, mathematically, coupled by interface conditions. These interface conditions express continuity of velocity and total stress. The fluids have different density $\rho$ and viscosity $\mu$, and surface tension forces act at the interface. In other words, the material properties are discontinuous. From a numerical point of view, an important simplification of the interface conditions is achieved by regularizing the viscosity. Similarly, the interface forces are mollified and effectively distributed to a bounded region around the interface.

## 2.2 Governing equations

Consider two fluids labeled '0' and '1' in a domain $\Omega \in \mathbb{R}^3$ which are separated by an interface $S$. Both fluids are assumed to be incompressible, i.e.:

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

where $\mathbf{u} = (u, v, w)^t$ is the velocity vector. The flow is governed by the incompressible Navier-Stokes equations:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \mu \left( \nabla \mathbf{u} + \nabla \mathbf{u}^t \right) + \mathbf{g}, \tag{2.2}$$

where $\rho$, $p$, $\mu$ and **g** are the density, pressure, viscosity and gravity vector respectively. The density and viscosity are constant within each fluid. When a

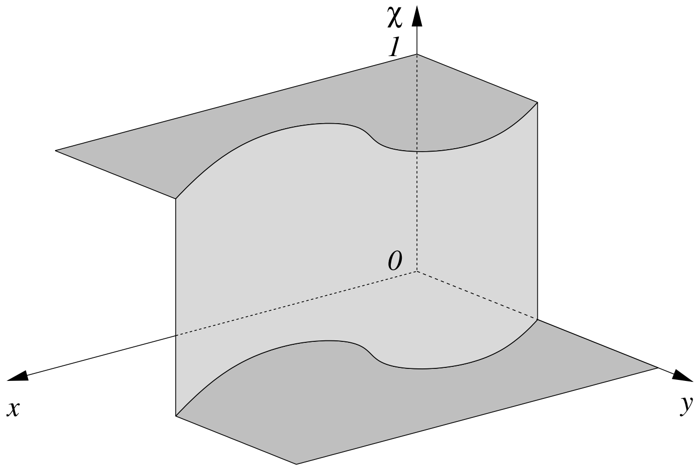


Figure 2.1: Color function $\chi$

color function $\chi$ is introduced as (see Fig. 2.1 for an example)

$$\chi = \begin{cases} 0, & \text{fluid `0'}, \\ 1, & \text{fluid `1'}, \end{cases} \tag{2.3}$$

then the density can be expressed as

$$\rho = \rho_0 + (\rho_1 - \rho_0)\chi \tag{2.4}$$

and the viscosity as

$$\mu = \mu_0 + (\mu_1 - \mu_0)\chi. \tag{2.5}$$

## 2.3 Surface tension

Cohesive forces act between the fluid molecules. The molecules at the interface do not have the same type of neighboring molecules on all sides, which results in surface tension. In this research surface tension is modeled on a macroscopic level. Consider an interface element $A$ as depicted in Fig. 2.2. It is assumed that a constant surface tension $\sigma$ acts in a direction tangential to $A$ and perpendicular to a tangent of $\partial A$. The force $\mathbf{f}$ acting on $A$ is then

$$\mathbf{f} = \oint_{\partial A} \sigma \mathbf{t} \, \mathrm{d}l. \tag{2.6}$$

The unit vector $\mathbf{t}$ can be written as

$$\mathbf{t} = \mathbf{s} \times \mathbf{n}, \tag{2.7}$$

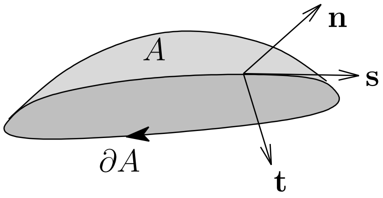


Figure 2.2: Surface tension acting on interface element $A$

where $\mathbf{s}$ is a unit vector tangential the the interface and $\mathbf{n}$ is a unit vector normal to $A$. Substitution yields

$$\mathbf{f} = \oint_{\partial A} \sigma \mathbf{s} \times \mathbf{n} \,\mathrm{d}l. \tag{2.8}$$

Application of the curl theorem results in

$$\mathbf{f} = \int_{A} \sigma (\mathbf{n} \times \nabla) \times \mathbf{n} \,\mathrm{d}A. \tag{2.9}$$

Assume that $\mathbf{n}$ is not only defined on the interface $A$ but anywhere in $\mathrm{I\!R}^3$. Then the following relation can be derived

$$(\mathbf{n} \times \nabla) \times \mathbf{n} = \nabla(\frac{1}{2}\|\mathbf{n}\|^2) - \mathbf{n}\nabla \cdot \mathbf{n}. \tag{2.10}$$

Substituting this in $\mathbf{f}$ and using $\|\mathbf{n}\| = 1$ finally shows that

$$\mathbf{f} = -\int_{A} \sigma \kappa \mathbf{n} \,\mathrm{d}A, \tag{2.11}$$

where $\kappa = \nabla \cdot \mathbf{n}$ is the mean curvature of the interface $A$. Obviously, surface tension causes a force normal to the interface of magnitude $\sigma\kappa$.

## 2.4 Interface boundary conditions

The interface boundary conditions express continuity of mass and momentum at the interface (which can be found in any relevant textbook, but see e.g. Tryggvason et al. [4] for a recent reference):

$$\begin{aligned} [\mathbf{u}] &= 0 \qquad (2.12) \\ \left[p\mathbf{n} + \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^t) \cdot \mathbf{n}\right] &= \sigma\kappa\mathbf{n} \qquad (2.13) \end{aligned}$$

where the brackets denote jumps across the interface, $\mathbf{n}$ is a normal vector at the interface, $\sigma$ is the surface tension coefficient and $\kappa$ is the curvature of the interface.

## 2.5 Regularization of viscosity

The interface conditions (2.13) show that the jump conditions for pressure and velocity are coupled. Furthermore, Li and Lai [5]) show that although the velocity is continuous at the interface, the velocity gradients are not. If $\mathbf{s}$ is a vector parallel to the interface, $u_n = \mathbf{n} \cdot \mathbf{u}$ is the interface normal velocity component and $u_s = \mathbf{s} \cdot \mathbf{u}$ a velocity component parallel to the interface, then

$$\begin{aligned} \left[\frac{\partial u_n}{\partial n}\right] &= 0, & \left[\frac{\partial u_n}{\partial s}\right] &= 0, \\ \left[\frac{\partial u_s}{\partial n}\right] &= -[\mu]\frac{\partial u_n}{\partial s}, & \left[\frac{\partial u_s}{\partial s}\right] &= 0. \end{aligned} \tag{2.14}$$

But note that if the viscosity $\mu$ is continuous at the interface, Eqn. (2.14) shows that the derivatives of the velocity components are continuous too. In that case Eqn. (2.13) reduces to

$$[\nabla \mathbf{u}] = \mathbf{0}, \tag{2.15}$$

$$[p] = \sigma\kappa. \tag{2.16}$$

This means that, besides that the velocity gradients are continuous at the interface, the jump conditions for pressure and velocity are decoupled, which are two major advantages for the numerical treatment of the moving boundary problem. Therefore the viscosity is made continuous by mollifying $\chi$ in expression (2.5), see Fig. 2.3:

$$\mu = \mu_0 + (\mu_1 - \mu_0)\chi_\alpha. \tag{2.17}$$

## 2.6 Continuous Surface Force (CSF)

The surface tension force contributes to a pressure jump $\sigma\kappa$ at the interface, as shown by Eqn. (2.16). This is numerically hard to treat, since the pressure becomes discontinuous. Instead of exactly taking into account the pressure jump at the interface due to the surface tension forces, the continuous surface force (CSF, Brackbill et al. [6]) methodology is adopted.

Two different ways in approaching multi-phase flows can be distinguished. The first considers each fluid or phase separately, having its own set of governing equations. The different fluids or phases are coupled by interface conditions. The other way is to look at the flow in whole and take into account the interface surface forces and (discontinuously) varying material constants. This is called the continuum approach. Eqn. (2.2) is then usually rewritten as:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \mu\left(\nabla \mathbf{u} + \nabla \mathbf{u}^t\right) + \mathbf{g} + \frac{1}{\rho}\mathbf{f}_s, \tag{2.18}$$

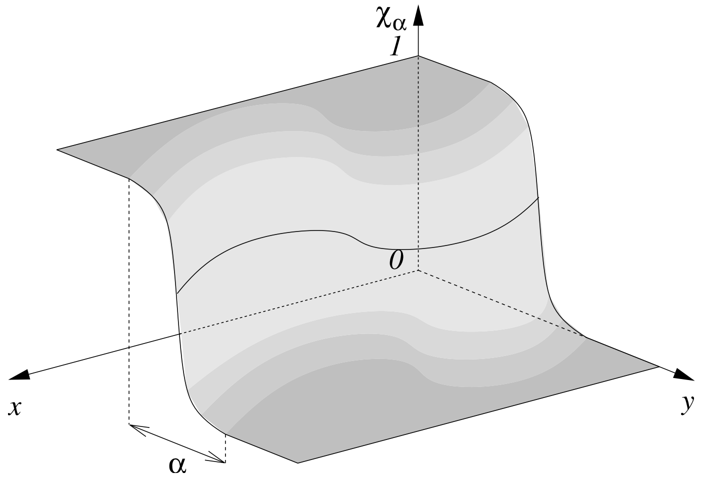


Figure 2.3: Regularized color function $\chi_\alpha$

where $\mathbf{f}_s$ is the surface tension force that acts at the interface only. Eqn. (2.18) now holds in the whole domain and the interface conditions follow from the proper prescription of $\mathbf{f}_s$. It is shown by Brackbill et al. [6] that the surface tension force can be expressed as:

$$\mathbf{f}_s = \frac{\rho}{\frac{1}{2}\left(\rho_0 + \rho_1\right)}\ \sigma\kappa\delta(d)\mathbf{n}, \tag{2.19}$$

where $\delta$ is the Dirac delta function, $d$ measures the nearest distance to the interface and $\mathbf{n}$ is the normal to the interface. The Dirac delta function illustrates that the surface tension force is a singular volume force.

The surface tension force can be formally rewritten as:

$$\mathbf{f}_s = \frac{\rho}{\frac{1}{2}\left(\rho_0 + \rho_1\right)}\ \sigma\kappa \lim_{\alpha\to 0} \nabla\chi_\alpha, \tag{2.20}$$

where $\alpha$ is the (vanishing) transition width of the mollified color function $\chi_\alpha$. The limit $\alpha \to 0$ illustrates that the surface tension force acts at the interface only. With the CSF approach, the surface tension forces are smeared out around the interface by keeping the transition width to some non-vanishing value, i.e. $\alpha > 0$:

$$\mathbf{f}_s = \frac{\rho}{\frac{1}{2}\left(\rho_0 + \rho_1\right)}\ \sigma\kappa\nabla\chi_\alpha. \tag{2.21}$$

This is equivalent to regularizing the Dirac delta function in Eqn. (2.20) (see Fig. 2.4):

$$\mathbf{f}_s = \frac{\rho}{\frac{1}{2}\left(\rho_0 + \rho_1\right)}\ \sigma\kappa\delta_\alpha(d)\mathbf{n}. \tag{2.22}$$

The consequence is that the pressure jump is now smeared out over the transi-

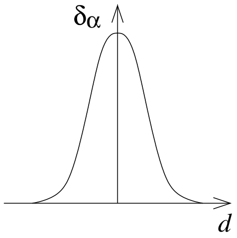


Figure 2.4: Regularized Dirac delta function $\delta_\alpha$

tion region $\alpha$. Since the pressure is forced to be continuous, the jump condition for the pressure becomes

$$[p] = 0. \tag{2.23}$$

The way the color function $\chi$ or the Dirac delta function $\delta$ are regularized varies for the different moving boundary methods. This also holds for the curvature $\kappa$ and interface normal $\mathbf{n}$.

## 2.7 Initial and Boundary conditions

In this work a rectangular domain is considered that is bounded by solid, rigid walls. Therefore no-slip boundary conditions are used for the velocity vector $\mathbf{u}$, i.e.:

$$\mathbf{u} = \mathbf{0}, \quad \text{at the wall} \tag{2.24}$$

and no boundary conditions have to be prescribed for the pressure $p$. Furthermore, the contact angle between the interface and the bounding wall has to be prescribed. A proper description of this angle requires the physical modeling of so-called wall adhesion (see e.g. Brackbill et al. [6]). It is not a material property of the fluid, but depends on for example the wall roughness. It becomes even more complicated when the interface is in motion. Since wall adhesion is not the focus of this research, it is assumed that the interface is perpendicular to the bounding walls.

The simulations presented in this work are started from rest with some prescribed interface position. This means that the initial condition for the velocity is $\mathbf{u} = \mathbf{0}$. No initial conditions are required for the pressure $p$. Note that the initial conditions are in agreement with the boundary conditions.

# Chapter 3

# Moving Boundary methods

## 3.1 Introduction

The mathematical model of the moving boundary problem has been presented in the previous chapter. There exist a moving, internal boundary in the domain, that is modeled as an interface. When the governing equations are discretized, the question remains how to deal with this interface.

The moving boundary problem can be formulated in a Lagrangian way, Eulerian way or a combination of both. The Lagrangian and combined formulations lead to moving, boundary conforming meshes. These are cumbersome for simulating large numbers of arbitrarily shaped interfaces. This technique is therefore not very suitable for the present work and will not be considered any further.

With an Eulerian formulation on the other hand (see Eqns. (2.1) and (2.2)), the computations are performed on a fixed mesh. Within the fixed mesh, an interface exists, where density and viscosity are discontinuous and surface tension forces act. Three fundamental issues are associated with the numerical treatment of the interface (see Shyy et al. [7]):

1. the discrete representation,
2. the evolution in time,
3. the manner in which interface boundary conditions are imposed.

Various methods have been put forward to treat moving boundary problems. A classification is presented in Table 3.1.

## 3.2 Capturing

With capturing methods the interface is seen as a contact discontinuity. The interface is defined in the initial conditions only. It is captured afterwards by techniques similar to shock-capturing methods for compressible flow. The

1. capturing methods
2. tracking methods
    - (a) front tracking
        - i. hybrid front-tracking front-capturing
        - ii. immersed boundary
        - iii. cut-cell
    - (b) volume tracking
        - i. marker and cell (MAC)
        - ii. volume of fluid (VOF)
            - A. Hirt-Nichol's VOF
            - B. simple line interface calculation (SLIC)
            - C. piecewise linear interface calculation (PLIC)
                - Eulerian interface advection
                - Lagrangian interface advection
            - D. Flux-Corrected Transport (FCT)
            - E. Constrained Interpolation Profile (CIP)
        - iii. Level-Set
        - iv. coupled Level-Set Volume-of-Fluid (CLSVOF)

Table 3.1: Classification of free-boundary methods

biggest drawback is the so-called convective averaging: convective flux calculation requires an averaging of the flow properties of all fluid elements in a given computational cell. This results in smoothing of variations and smearing of discontinuities. It is not considered any further.

## 3.3 Front tracking

With tracking methods, additional effort is made to explicitly locate the interface. The *front-tracking* method (Unverdi and Tryggvason [8], Tryggvason et al. [4]) and the closely related *immersed boundary* method (Lai and Peskin [9]) use a fixed mesh to solve the model equations, but use moving interface meshes to explicitly track the interface. The interface conditions are satisfied by regularizing (smoothing) the interface discontinuities and interpolating interface forces from the interface grid to the fixed grid. For this purpose, the interface forces are transformed into volume forces and distributed over a zone with non-zero width. This is sometimes referred to as the *continuous surface force* (CSF) approach (Brackbill et al. [6]). In the *cut-cell* approach (Udaykumar et al. [10], Ye et al [11]) on the other hand, the interface conditions are satisfied without smoothing of the interface.

## 3.4 Volume tracking

The interface grid will be difficult to evaluate when the interface has arbitrary shape and topology. Therefore the volume tracking methodology is preferred for the present research. Instead of tracking the interface, to every fluid a *color* is uniquely assigned. The change of color implicitly defines the interface. The advantage is that coalescence and changes in topology are automatically accounted for.

In case of the Marker-and-Cell method (Harlow and Welch [12]), the fluids are colored by marker particles. The particles are initially distributed in the fluids and subsequently advected by the fluid flow in a Lagrangian way (see Fig. 3.1):

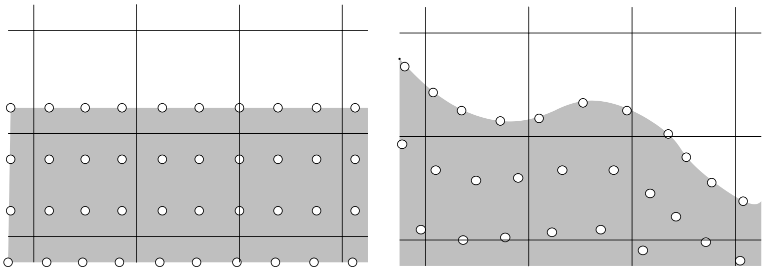
$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{u}(\mathbf{x}, t), \tag{3.1}$$

where $\mathbf{x}$ is the position of a particle and $\mathbf{u}$ is the velocity vector inside the flowfield. The method has been applied to free-surface flows in [12], where pressure boundary conditions are specified at the free surface.

In the Volume-of-Fluid and Level-Set methods, a coloring *function*, say $\Phi$ is used to identify the separate fluids. The function is (in principle) advected in an Eulerian way:

$$\frac{\partial \Phi}{\partial t} + \nabla \cdot (\Phi \mathbf{u}) = 0. \tag{3.2}$$

With the Volume-of-Fluid method, the coloring function measures the fractional volume of a certain fluid within a computational cell. It has a step-like behavior. The major advantage of VOF methods is that the masses of each fluid



(a) initial distribution (b) instantaneous distribution

Figure 3.1: Marker particles; Marker-and-Cell method

are conserved, provided the discretization is conservative. On the other hand, the step-like behavior of the coloring function makes the advection elaborate. Besides that, interface normals and curvature are not easily computed.

With the Level-Set method, the interface is defined by the zero-level set of a coloring function. It is chosen such that it is smooth near the interface. Methods available from hyperbolic conservation laws can be used to advect the interface. When the coloring function is sufficiently smooth, the interface normal directions and curvature can easily be computed. The major disadvantage is that it is not rigorously mass-conserving. This means that additional effort is necessary to conserve mass, or at least to improve mass conservation. The CLSVOF method combines the Level-Set method with a Volume-of-Fluid method (Sussman and Puckett [13, 14]) to improve mass conservation. The Hybrid Particle Level Set Method ([15, 16]) uses additional marker particles.

### 3.4.1 Volume-of-Fluid method

We will give a brief description of the Volume-of-Fluid method. For more detailed descriptions see references [17–33]. The Volume-of-Fluid method is a volume-tracking method. The interface is implicitly defined by the Volume-of-Fluid function $\Psi$. This function measures the fractional volume of a certain fluid in a computational cell. The Volume-of-Fluid function has a step-like behavior: it can be zero, one or somewhere in between if the computational cell contains the interface. These cells are called mixed cells.

Following Hirt and Nichols [19], Popinet and Zaleski [21], Gueyffier et al. [23] and Rider and Kothe [24], a characteristic function $\chi$ is introduced. It has value 0 in fluid '0' and 1 in fluid '1'. The Volume-of-Fluid function is therefore:

$$\Psi(\mathbf{x}_k) = \frac{1}{\mathrm{vol}(\Omega_k)} \int\limits_{\Omega_k} \chi \, \mathrm{d}\Omega, \tag{3.3}$$

where $\Omega_k$ is a computational cell and $\mathbf{x}_k$ the corresponding node. To find an evolution equation for $\Psi$, first consider a fluid particle that moves with the

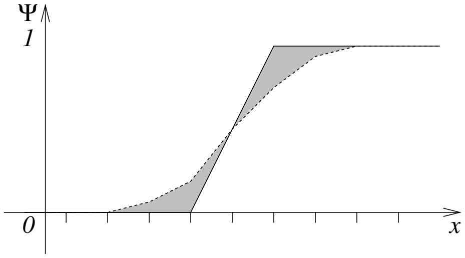


Figure 3.2: One-dimensional example of fractional volume $\Psi$; loss of interface definition after time step

flow-field:

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{u}(\mathbf{x}, t). \tag{3.4}$$

Since the particle does not cross the interface, the value of $\chi$ of the particle remains constant:

$$\chi(\mathbf{x}(t), t) = \text{constant}, \tag{3.5}$$

from which follows

$$\frac{\partial \chi}{\partial t} + \mathbf{u} \cdot \nabla \chi = 0. \tag{3.6}$$

From Eqn. (3.3) follows:

$$\frac{\mathrm{d}\Psi(\mathbf{x}_k)}{\mathrm{d}t} = \frac{1}{\mathrm{vol}(\Omega_k)} \int\limits_{\Omega_k} \frac{\mathrm{d}\chi}{\mathrm{d}t} \,\mathrm{d}\Omega = -\frac{1}{\mathrm{vol}(\Omega_k)} \int\limits_{\Omega_k} \mathbf{u} \cdot \nabla \chi \,\mathrm{d}\Omega. \tag{3.7}$$

If we use $\nabla \cdot \mathbf{u} = 0$:

$$\frac{\mathrm{d}\Psi(\mathbf{x}_k)}{\mathrm{d}t} = -\frac{1}{\mathrm{vol}(\Omega_k)} \int\limits_{\Omega_k} \nabla \cdot (\chi \mathbf{u}) \,\mathrm{d}\Omega = -\frac{1}{\mathrm{vol}(\Omega_k)} \int\limits_{\partial\Omega_k} \chi \mathbf{u} \cdot \mathbf{n} \,\mathrm{d}S. \tag{3.8}$$

Mixed cells ($0 < \Psi < 1$) may only exist when the interface intersects the computational cell (step-like behavior). In that sense the Volume-of-Fluid function is not allowed to smear out, it has to remain sharp near the interface. This is *required* to maintain a good interface representation (Rudman [25]). So special care has to be taken when Eqn. (3.8) is discretized. As an example, Fig. 3.2 illustrates a careless application of a flux approximation scheme. Due to numerical diffusion, there are multiple mixed cells and the interface becomes ill-defined.

Maintaining a sharp interface is basically achieved in two essentially different ways. First of all, the fluxes in Eqn. (3.8) can be approximated by adopting *algebraic* methods which preserve the discontinuity of $\Psi$. The Flux-Corrected-Transport (FCT, Boris and Book [34], Zalesak [17], Rudman [25]) method serves this goal. The Constrained Interpolation Profile (CIP, Yabe et al. [35]) method can also be used as an algebraic advection procedure.

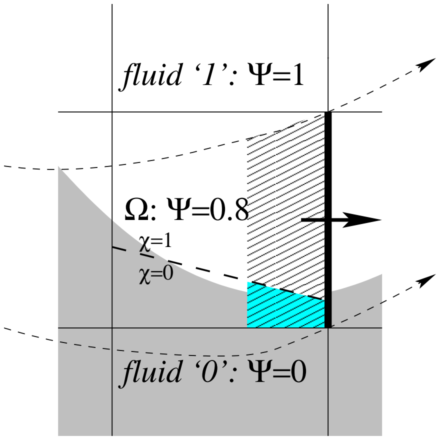


Figure 3.3: Donating region (hatched) defined by instantaneous streamlines (dashed) for right-hand-side face of a mixed cell

Alternatively, the fluxes can be approximated in a geometric way. Firstly, the interface is *reconstructed* within a mixed cell from the distribution of $\Psi$, see Fig. 3.3. This could then be seen as an approximation for $\chi$ in the mixed cell, which has value 0 on one side of the interface and 1 on the other. The fluxes are computed in a geometrical manner. Within a computational cell, a *donating* region is defined, which is the part of the mixed cell containing fluid that will be fluxed through a certain cell boundary (face) during a period of time (time step), see Fig. 3.3. Assuming constant flow during a period of time $\Delta t$ (time step), fluid enclosed by the limiting stream-lines will eventually flow through the face. The intersection of the donating region with the interface divides the total amount of fluxed fluid into a contribution from 'fluid 0' ($\chi = 0$ ) and 'fluid 1' ($\chi = 1$). Note that each part of the boundary of a computational cell (face) has its own donating region.

The latter methods, based on a geometric flux computation, can be further classified according to the interface approximation employed (see Table 3.1). In case of 'simple line interface calculation' (SLIC, Noh and Woodward [18], ) and 'Hirt-Nichol's VOF' method (Hirt and Nichols [19]), the interface is assumed to be aligned to a coordinate direction. 'Hirt-Nichol's VOF' also allows stair-stepped interfaces within a mixed cell. For these two methods, the volume fluxes can be expressed algebraically, without the need of interface reconstruction. In case of 'piecewise linear interface calculation' methods (PLIC, see References [22–26, 29–33]), the interface inside a mixed cell is approximated by a line in two dimensions or plane surface in three dimensions. Unlike Hirt-Nichol's VOF and SLIC, the interface can acquire any orientation in a mixed cell with PLIC.

The methodology of PLIC is indicated in Table 3.2. By estimation of a normal direction to the interface inside a computational cell (stage 1), the interface can be reconstructed (stage 2). The volume fraction $\Psi$ is updated by advecting

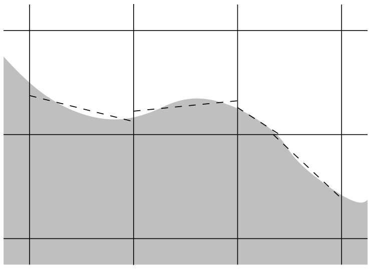


Figure 3.4: Interface reconstruction; VOF/PLIC

1. normal estimation
2. interface reconstruction
3. advection of interface
    - Eulerian
    - Lagrangian

Table 3.2: Methodology of PLIC

the interface (stage 3). This is achieved in either an Eulerian or a Lagrangian way. From the distribution of $\Psi$, a normal direction to the interface and the fractional volume of fluid '0' in a computational cell can be derived. At each time step a planar surface is constructed having the same normal direction and dividing the cell into two parts, so that it contains the proper amount of fluids '0' and '1'. Having reconstructed the interface (stages 1 and 2 of Table 3.2), the interface has to be advected in order to update the volume fraction $\Psi$. In case of an Eulerian point of view, the fluxes of fluids '0' (and '1') over the boundaries of a computational cell

$$\frac{1}{\mathrm{vol}(\Omega_k)} \int\limits_{\Delta t} \int\limits_{\partial\Omega_k} \chi \mathbf{u} \cdot \mathbf{n} \, \mathrm{d}S \ \mathrm{d}t$$

are calculated similar to the donor-acceptor method. This is typically achieved in a geometrical manner (Rider and Kothe [24], Rudman [25], Harvie and Fletcher [28, 29], Lafaurie et al. [20]). In case of Lagrangian interface advection, the interface position is updated by moving its corners by an amount 'velocity times time step' (Renardy et al. [22], Gueyffier [23]).

### 3.4.2 Level-Set method

An alternative to the Volume-of-Fluid methods is the Level-Set method (Chang et al. [36], Sussman et al. [37], Iafrati et al. [38], Mulder et al. [39], Zhu and

Sethian [40], Sethian [41, 42], Osher and Fedkiw [43], Sussman et al. [1, 44, 45], Zheng and Zhang [46], Kaliakatsos and Tsangaris [47]). It is, like the Volume-of-Fluid-method, a volume tracking method. In other words, the interface is implicitly defined by a marker function. A marker function $\Phi$ is chosen such that it changes sign at the interface. The interface is the zero level-set of $\Phi$ :

$$S(t) = \{\mathbf{x} | \Phi(\mathbf{x}, t) = 0\}. \tag{3.9}$$

When a superscript $^+$ or $^-$ corresponds to a positive or negative signed distance $\Phi$ from the interface respectively, the density $\rho$ can be expressed as:

$$\rho = \rho^- + H(\Phi)(\rho^+ - \rho^-), \tag{3.10}$$

and similarly for the viscosity $\mu$, where $H(\Phi)$ is the Heaviside step function.

The interface is evolved by advecting the Level-Set function in the flow field as if it were a material property:

$$\frac{\partial \Phi}{\partial t} + \mathbf{u} \cdot \nabla \Phi = 0. \tag{3.11}$$

The Level-Set function $\Phi$ is typically a smooth function that, unlike the Volume-of-Fluid method, allows the straightforward application of a difference scheme. Due to the implicit definition of the interface, arbitrary topologies, merging and break-up are automatically dealt with. Furthermore, it is easily generalized to three dimensions.

Homogeneous Neumann boundary conditions are applied for the Level-Set function at the solid walls. This means that the interface is assumed to be perpendicular to the wall where the interface intersects with the bounding wall. An appropriate initial choice for $\Phi$ is a distance function, which measures the (signed) distance to the nearest interface surface. It is important to note that the Level-Set function does not remain a distance function when it is advected in the course of time.

The geometrical properties of the interface are easily determined from the Level-Set function. The unit normal to an iso-contour of the level-set function is given by:

$$\mathbf{n} = \frac{\nabla \Phi}{|\nabla \Phi|} \tag{3.12}$$

The mean curvature $\kappa$ is defined by

$$\kappa = \nabla \cdot \mathbf{n} = \nabla \cdot \frac{\nabla \Phi}{|\nabla \Phi|}. \tag{3.13}$$

The major drawback of the Level-Set method is that although the Level-Set function itself might be conserved, this does not imply that the area enclosed by an interface is conserved. Fig. 3.5 shows a one-dimensional example. Interfaces are prescribed at two different $x$-locations. Starting from a distance function, time is advanced by one time step. Convective smoothing will essentially smooth

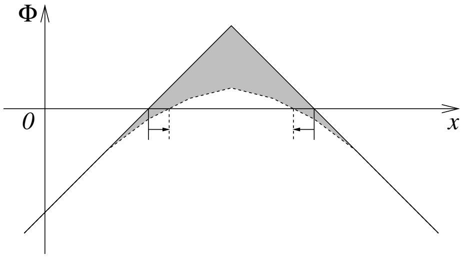


Figure 3.5: One-dimensional example of the level-set function Φ; mass errors after time step

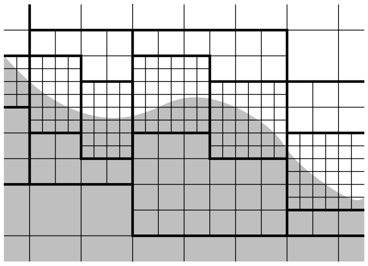


Figure 3.6: Adaptive grid refinement

function Φ. As seen in Fig. 3.5, this (once again, artificial) smoothing shifts the interface and causes an enclosed region of fluid to loose mass.

Area preservation can be improved by higher order discretization. The Essentially Non-oscillatory (ENO) scheme serves this goal (Shu and Osher [48], Harten et al. [49, 50]). Time is evolved in [37] by adopting the second order Adams-Bashforth method . Improvement is observed by employing the Runge-Kutta scheme in [44], where third order Runge-Kutta is adopted in combination with third order ENO.

Efficiency of the method can be improved by adopting the so-called Narrow Band Level-Set method (Sethian [41, 42]). The computational effort is decreased by performing the calculations only in a narrow band of the zero level-set. Furthermore, the mesh can be adaptively refined locally near the interface (Haj-Hariri and Shi [51], Sussman et al. [1], Strain [52]), see Fig. 3.6.

### Reinitialization

If an initial signed distance function is advected through a non-uniform flow, function Φ does not necessarily correspond to a distance function any longer. A distance function satisfies

$$|\nabla\Phi| = 1. \tag{3.14}$$

If, however, if it is required that the Level-Set function $\Phi$ remains a signed normal distance function at all instances (for reasons given later), function $\Phi$ can be reinitialized each time step. Sussman et al. [37] use a partial differential equation to keep $\Phi$ a distance function:

$$\begin{aligned} \frac{\partial\Phi}{\partial t'} &= \operatorname{sign}(\Phi_0)\,(1-|\nabla\Phi|)\,, \\ \Phi_0 &= \Phi|_{t'=0}\,, \end{aligned} \tag{3.15}$$

where $t'$ is an artificial time. The sign function causes the zero level-set to be left unchanged, so that reinitializing does not result in repositioning of the interface.

A numerical procedure for the reinitialization is given by Sethian in [41]. It is based on the (one-dimensional) equivalence with hyperbolic conservation laws. The ENO scheme is used for higher order spacial discretization. For the temporal discretization the Adams-Bashfort and Runge-Kutta methods are applied in [44, 45].

According to Chang et al. [36] and Zheng and Zhang[46], the reinitialization procedure causes a considerable amount of additional mass losses. Numerical diffusion introduces a normal motion of the interface proportional to the interface's local curvature. Increasing the order of discretization reduces the area errors. The reinitialization procedure has been modified by Chang et al. [36], Zheng and Zhang [46], Sussman et al. [1, 44, 45] and Peng et al. [53] to improve area preservation of re-initialization.

## 3.5 Combined methods

Various approaches exist that combine the methods discussed above and are intended to overcome the specific problems associated with the separate methods. An indicator function is used by Shin et al. [54, 55] and De Sousa et al. [2] to deal with topology changes with the front tracking method and improve the computation of curvature of the interface. Ceniceros and Roma [56] use the Level-Set methodology to solve the flow field, but use a front tracking technique to advect the interface, without Level-Set advection and re-initialization.

Enright et al. [15, 16] improve the mass-conservation of the Level-Set method by adding passively advected marker particles. These particles are used near the interface. The *Coupled Level-Set Volume-of-Fluid* (CLSVOF) method of Sussman and Puckett [13, 14]) is a coupling of the Level-Set method with the Volume-of-Fluid PLIC method. Besides the advection of the Level-Set function $\Phi$ also the Volume-of-Fluid function $\Psi$ is advected. There is no straightforward relationship between the Level-Set function and the Volume-of-Fluid function; both advections are independent. After each update of $\Phi$ and $\Psi$, coupling of both functions takes place. This coupling is not easily achieved. Since the PLIC approach is employed, a drawback of this method might be that besides mass-conservation, also the elaborateness of the VOF methods is imported. The mass-conservation properties are shown to be comparable to VOF methods.

## 3.6 Conclusion

The two methods that are of most interest are the Volume-of-Fluid method and the Level-Set method. The advantage of the Volume-of-Fluid method is that it is rigorously mass conserving. On the other hand, much attention has to be paid to the advection of a discontinuous marker function. For that various procedures exist. The piecewise linear interface calculation is superior to the others. However, the interface has to be reconstructed each time step, which certainly is an elaborate task, especially in three-dimensional space.

Another drawback of the Volume-of-Fluid method is that due to the discontinuous nature of the marker function, the interface curvature is not easily determined. The curvature is essential for modeling surface tension effects. With the Level-Set method, on the other hand, the surface curvature is very easily determined, since the Level-Set function is smooth near the interface. However, it is not mass conserving. Also, advecting the interface is possible by application of 'of-the-shelf' techniques for hyperbolic conservation laws. For these reasons, the Level-Set method has been chosen as the basis of this work. However, mass-conservation is not an intrinsic property and this is considered the major drawback of the Level-Set method.

The focus of the present research is on a mass-conserving way to advect the interface. Our work has a shared foundation with the CLSVOF method (Sussman and Puckett [13, 14]) and to a lesser extent with the combined Level-Set/particle method (Enright et al. [15, 16]) in the sense that it is based on Level-Set and additional effort is made to conserve mass. The difference between our method and CLSVOF is that there is no combination of two existing methods. The method takes full advantage from all additional information provided by the Level-Set function $\Phi$, rather than coupling Level-Set with Volume-of-Fluid/PLIC. In fact, the Volume-of-Fluid function $\Psi$ is used only to conserve mass, without applying the difficult convection step (namely interface reconstruction) which makes VOF so elaborate.

An explicit relation between the Level-Set function $\Phi$ and Volume-of-Fluid function $\Psi$ is proposed, which is the basis of this work. This relation is obtained by assuming piecewise linear interfaces within a computational cell, and can be written as:

$$\Psi = f(\Phi, \nabla\Phi). \tag{3.16}$$

This relationship makes advection of the Volume-of-Fluid function $\Psi$ easy (i.e. without interface reconstruction) and finding $\Phi$ from $\Psi$ is straightforward. The PLIC method is not used (unlike CLSVOF), yet mass is conserved in the same manner. Note that the PLIC method (and consequently the CLSVOF method) might not be easily extendible to 3D. Extension of our method to 3D is straightforward. Note also that with this approach, it is not necessary to smooth (or regularize) $\Psi$, which is usually necessary in other methods.

# Chapter 4

# Numerical Approach

## 4.1 Introduction

The fluid flow is described by the continuity equation (see Eqn. (2.1)):

$$\nabla \cdot \mathbf{u} = 0, \tag{4.1}$$

and the Navier-Stokes equations (see Eqn. (2.18)):

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^t\right) + \mathbf{g} + \frac{1}{\rho}\mathbf{f}_s. \tag{4.2}$$

The equations are solved on a Cartesian grid in a rectangular domain by the pressure-correction method ([57]). The unknowns are stored in a Marker-and-Cell (staggered) layout ([12]). For the interface representation the Level-Set methodology is adopted. The interface conditions are satisfied by means of the continuous surface force (CSF) methodology. The discontinuous density field is dealt with similarly to the ghost fluid method for incompressible flow ([58]).

For the interface representation and advection the Level-Set approach is used. The interface advection is decoupled from the flow-field computations. That means that the flow equations (Eqns. (4.1) and (4.2)) are solved with a given interface position. In this chapter the discretization of these flow equations is considered. The advection of the interface is discussed in Chapter 5.

## 4.2 Regularization of viscosity

The viscosity is regularized as explained in Section 2.5. It is expressed as (see Eqn. (2.17)):

$$\mu = \mu_0 + (\mu_1 - \mu_0)\chi_\alpha, \tag{4.3}$$

where $\mu_0$ is the viscosity of fluid '0', $\mu_1$ is the viscosity of fluid '1' and $\chi_\alpha$ is the is a mollified characteristic function. In the Level-Set approach, the characteristic function is a regularized Heaviside step function $H_\alpha$ of $\Phi$, i.e.:

$$\chi_\alpha = H_\alpha(\Phi). \tag{4.4}$$

Here $H_\alpha$ is chosen as (see e.g. Sussman et al. [37]):

$$H_\alpha(\Phi) = \begin{cases} 0 & \Phi < -\alpha, \\ \frac{1}{2}\left(1+\sin(\frac{x}{\alpha}\frac{1}{2}\pi)\right) & |\Phi| \leq \alpha, \\ 1 & \Phi > \alpha, \end{cases} \tag{4.5}$$

and $\alpha$ is a parameter proportional to the mesh width $h$. Following Sussman et al. [37]) $\alpha$ is chosen as $\alpha = \frac{3}{2}h$. According to Chang et al. [36], the viscosity is then smoothed over three mesh widths, provided $|\nabla\Phi| = 1$. Note that only the viscosity is smoothed, not the density $\rho$. Note also that when the density is not regularized, mass is conserved when the *volume* of a certain fluid or phase is conserved. In fact, our method conserves volumes by construction. Due to the non-regularized density-field, mass is conserved too.

## 4.3 Continuous Surface Force

In Eqn. (4.2) the surface tension force $\frac{1}{\rho}\mathbf{f}_s$ appears. In Section 2.6 it is expressed as (see Eqn. (2.22)):

$$\frac{1}{\rho}\mathbf{f}_s = \frac{1}{\frac{1}{2}(\rho_0+\rho_1)}\,\sigma\kappa\delta_\alpha(d)\mathbf{n}, \tag{4.6}$$

where $\delta_\alpha$ is a regularized Dirac delta function, $d$ is the nearest distance to the interface, $\rho_0$ is the density of fluid ‘0’, $\rho_1$ the density of fluid ‘1’, $\sigma$ is the surface tension coefficient and $\kappa$ is the curvature of the interface. In the Level-Set methodology, the normal $\mathbf{n}$ is given by Eqn. (3.12) and $\delta_\alpha(d) = \delta_\alpha(\Phi)|\nabla\Phi|$, see Chang et al. [36]. The surface tension force is then:

$$\mathbf{f}_s = \frac{\rho}{\frac{1}{2}(\rho_0+\rho_1)}\,\sigma\kappa\delta_\alpha(\Phi)\nabla\Phi. \tag{4.7}$$

The curvature of the interface is is given by Eqn. (3.13):

$$\kappa = \nabla\cdot\frac{\nabla\Phi}{|\nabla\Phi|}, \tag{4.8}$$

which is approximated by central differences. The Dirac delta function is regularized in the same manner as the Heaviside step function (Eqn. (4.5))

$$\delta_\alpha(\Phi) = \begin{cases} \frac{1}{2\alpha}\left(1+\cos(\frac{\Phi}{\alpha}\pi)\right) & |\Phi| \leq \alpha, \\ 0 & |\Phi| > \alpha. \end{cases} \tag{4.9}$$

Here $\alpha$ has the same value as in Eqn. (4.5), i.e. $\alpha = \frac{3}{2}h$. Note that the density $\rho$ is discontinuous. To guarantee a straightforward application of the pressure-correction methodology it is necessary that $\frac{1}{\rho}\mathbf{f}_s$ is regular. This is achieved since $\frac{1}{\rho}\mathbf{f}_s$ is regularized and not $\mathbf{f}_s$.

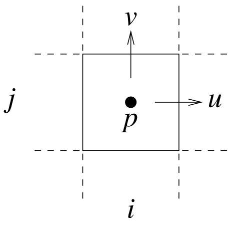


Figure 4.1: Grid layout in two dimensions

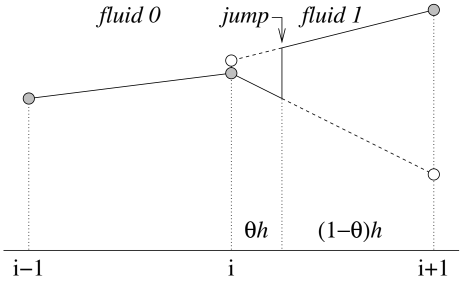


Figure 4.2: Ghost Fluid method; real values (gray) and ghost values (white)

## 4.4 Spatial discretization

Finite differences are used for the spatial discretization of Eqns. (4.1) and (4.2). The unknowns $u$, $v$, $w$ and $p$ are stored in a staggered way, as indicated in Fig. 4.1 for two dimensions. Away from the interface, the discretization is performed by straightforward central differencing. Velocities which have to be stored and evaluated at different locations, are approximated by averaging (following Harlow and Welch [12]). Near the interface, the jump conditions have to be taken into account in the discretization of the gradients. For that purpose the Ghost-Fluid method (Liu et al. [59]) is applied.

With the Ghost Fluid Method, the fluid at one side of the interface is continuously extended towards the other side. These values are called ghost values, Fig. 4.2. The ghost values are obtained by employing the previously mentioned jump conditions. Consider flow quantity $p$ (not necessarily the pressure). The goal is to compute derivatives $\frac{\partial p}{\partial x}$, $\frac{\partial p}{\partial y}$ and $\frac{\partial p}{\partial z}$. Quantity $p$ has *known* jumps $[p]$, $[\beta\frac{\partial p}{\partial x}]$, $[\beta\frac{\partial p}{\partial y}]$ and $[\beta\frac{\partial p}{\partial z}]$. Appearing in these jumps is $\beta$, which is a material constant ($\frac{1}{\rho}$ or $\mu$ or $\nu$) and assumed to be constant within each separate fluid or phase. Assume that jumps exist between $i$ and $i+1$ and that $i$ is within fluid or phase 0 and, consequently, $i+1$ is within fluid or phase 1. The jumps are discretized as:

$$\begin{array}{rcl} p_1 - p_0 & = & [p] \\ \beta_1\frac{\partial p_1}{\partial x} - \beta_0\frac{\partial p_0}{\partial x} & = & [\beta\frac{\partial p}{\partial x}]. \end{array} \tag{4.10}$$

The interface values $p_0|_{\mathrm{I}}$ and $p_1|_{\mathrm{I}}$ are eliminated from the (approximated) jump conditions:

$$\begin{array}{rcl} p_1|_{\mathrm{I}} - p_0|_{\mathrm{I}} & = & [p] \\ \beta_1 \frac{p_{i+1}-p_1|_{\mathrm{I}}}{(1-\theta)h} - \beta_0 \frac{p_0|_{\mathrm{I}}-p_i}{\theta h} & = & [\beta \frac{\partial p}{\partial x}], \end{array} \tag{4.11}$$

where $\theta$ is defined according to Fig. 4.2. This yields

$$\begin{array}{rcl} p_0|_I & = & \frac{\theta\beta_1 p_{i+1}+(1-\theta)\beta_0 p_i-\theta\beta_1[p]-\theta(1-\theta)h[\beta\frac{\partial p}{\partial x}]}{\theta\beta_1+(1-\theta)\beta_0} \\ p_1|_I & = & \frac{\theta\beta_1 p_{i+1}+(1-\theta)\beta_0 p_i+(1-\theta)\beta_1[p]-\theta(1-\theta)h[\beta\frac{\partial p}{\partial x}]}{\theta\beta_1+(1-\theta)\beta_0}. \end{array} \tag{4.12}$$

By linear extrapolation, the ghost values can be found. However, this is shortcut by finally expressing

$$\beta_0 \left.\frac{\partial p_0}{\partial x}\right|_{i+\frac{1}{2}} = \frac{p_0|_{i+1} - p_i}{h} = \frac{p_0|_I - p_i}{\theta h}, \tag{4.13}$$

which is

$$\beta_0 \left.\frac{\partial p_0}{\partial x}\right|_{i+\frac{1}{2}} = \hat{\beta}\frac{p_{i+1} - p_i}{h} - \hat{\beta}\frac{[p]}{h} - \frac{\hat{\beta}}{\beta_1}(1-\theta)[\beta\tfrac{\partial p}{\partial x}], \tag{4.14}$$

where $\hat{\beta}$ is the weighted harmonic average

$$\hat{\beta} = \frac{\beta_0\beta_1}{\theta\beta_1 + (1-\theta)\beta_0}. \tag{4.15}$$

After some bookkeeping, $\beta\frac{\partial p}{\partial x}$ is obtained as

$$\left(\beta\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2}} = \begin{cases} \hat{\beta}\left(\frac{p_{i+1}-p_i-[p]}{\Delta x} - \frac{1-\theta}{\beta_{i+1}}[\beta\frac{\partial p}{\partial x}]\right), & \text{left,} \\ \hat{\beta}\left(\frac{p_{i+1}-p_i-[p]}{\Delta x} + \frac{\theta}{\beta_i}[\beta\frac{\partial p}{\partial x}]\right), & \text{right,} \end{cases} \tag{4.16}$$

where by left and right the derivatives left and right from the interface are meant. In this expression $\theta$ marks the interface position. In case of the Level-Set methodology it is approximated by finding $\Phi = 0$ from a linearization of $\Phi$ between nodes $i$ and $i+1$:

$$\Phi_i + \theta(\Phi_{i+1} - \Phi_i) = 0. \tag{4.17}$$

This results in

$$\theta = \frac{-\Phi_i}{\Phi_{i+1} - \Phi_i}. \tag{4.18}$$

Furthermore, $\hat{\beta}$ is the weighted harmonic average of $\beta$

$$\hat{\beta} = \frac{\beta_{i+1}\beta_i}{\beta_{i+1}\theta + \beta_i(1-\theta)}. \tag{4.19}$$

Note that it is assumed, that the jump conditions can be computed at *all* locations $i$. If such a jump condition would depend on, for instance, the interface curvature, the curvature needs to be evaluated at places away from the interface.

Fortunately, the Level-Set methodology can serve this goal. The expressions for the flux $\beta\frac{\partial p}{\partial y}$ and $\beta\frac{\partial p}{\partial z}$ are analogous. As a matter of fact, it is easily obtained by switching the indices and replacing $\Delta x$ by $\Delta y$ and $\Delta z$ respectively.

Eqn. (4.16) is used for the discretization of the velocity gradient $\nabla u$ and the pressure gradient $\nabla p$. Due to the regularization of the viscosity, the velocity vector and its gradients are continuous at the interface (see Eqns. (2.12) and (2.15)). Therefore no special measures have to be taken for the discretization of the terms $\nabla \mathbf{u}$ near the interface:

$$\left(\frac{\partial u}{\partial x}\right)_{i+\frac{1}{2}} = \frac{u_{i+1} - u_i}{\Delta x}. \tag{4.20}$$

With this discretization of the first order derivative, second order derivatives are constructed in the usual way, for example:

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_i = \frac{\left(\frac{\partial u}{\partial x}\right)_{i+\frac{1}{2}} - \left(\frac{\partial u}{\partial x}\right)_{i-\frac{1}{2}}}{\Delta x}. \tag{4.21}$$

The pressure is continuous due to the application of the CSF method (see Eqn. (2.23)). The pressure gradient $\nabla p$ on the other hand is not. But, following Kang et al. [58], the jump of $\frac{1}{\rho}\nabla p$ is

$$[\frac{1}{\rho}\nabla p] = \mathbf{0}. \tag{4.22}$$

The discretization of the pressure gradient is therefore:

$$\left(\frac{1}{\rho}\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2}} = \frac{1}{\theta\rho_i + (1-\theta)\rho_{i+1}}\frac{p_{i+1} - p_i}{\Delta x}, \tag{4.23}$$

and similar in the other coordinate directions.

## 4.5 Temporal discretization

The flow-field is computed with a given interface position and the interface is advected with the new flow-field. Symbolically, if

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} &= \mathcal{A}(\mathbf{u}, \Phi), \\ \frac{\partial \Phi}{\partial t} &= \mathcal{B}(\Phi, \mathbf{u}), \end{aligned} \tag{4.24}$$

then the decoupled temporal discretization is given by

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = A(\mathbf{u}^n, \mathbf{u}^{n+1}, \Phi^{n+\frac{1}{2}}), \tag{4.25}$$

$$\frac{\Phi^{n+\frac{3}{2}} - \Phi^{n+\frac{1}{2}}}{\Delta t} = B(\Phi^{n+\frac{1}{2}}, \Phi^{n+\frac{3}{2}}, \mathbf{u}^{n+1}), \tag{4.26}$$

where $A$ and $B$ are the discrete counterparts of operators $\mathcal{A}$ and $\mathcal{B}$ respectively and superscript $^n$ denotes time-level $n$. Note that the Level-Set function $\Phi$ is staggered in time to make the decoupling formal second order accurate. However, the accuracy depends of course on the discretization of Eqns. (4.25) and (4.26), which in this research is first order. Note also that due to the regularization of viscosity and surface tension, higher order accuracy might not be expected.

The interface advection (Eqn. (4.26)) is discussed in detail in Chapter 5. For the time integration of the flow equations (Eqn. (4.25)) it is important to note that the Level-Set function at time $t^{n+\frac{1}{2}}$ is used. This means that the viscosity $\mu$ is defined at that time-instant:

$$\mu = \mu(\Phi^{n+\frac{1}{2}}). \tag{4.27}$$

The governing equations after spatial discretization are:

$$\left(\frac{\partial u}{\partial x}\right)_h + \left(\frac{\partial v}{\partial y}\right)_h + \left(\frac{\partial w}{\partial z}\right)_h = 0 \tag{4.28}$$

and

$$\begin{aligned} \frac{\mathrm{d}u_h}{\mathrm{d}t} = -u_h\left(\frac{\partial u}{\partial x}\right)_h - v_h\left(\frac{\partial u}{\partial y}\right)_h - w_h\left(\frac{\partial u}{\partial z}\right)_h - \frac{1}{\rho}\left(\frac{\partial p}{\partial x}\right)_h + \\ \tfrac{1}{\rho}\left(\frac{\partial\mu\frac{\partial u}{\partial x}}{\partial x} + \frac{\partial\mu\frac{\partial u}{\partial y}}{\partial y} + \frac{\partial\mu\frac{\partial u}{\partial z}}{\partial z} + \frac{\partial\mu\frac{\partial u}{\partial x}}{\partial x} + \frac{\partial\mu\frac{\partial v}{\partial x}}{\partial y} + \frac{\partial\mu\frac{\partial w}{\partial x}}{\partial z}\right)_h + \\ g_x + \tfrac{1}{\rho}\left(f_{s\,x}\right)_h, \end{aligned} \tag{4.29}$$

where subscript $_h$ indicates the discrete approximation, $g_x$ is the component of the gravity vector $\mathbf{g}$ in $x$-direction and $f_{s\,x}$ is the component of the surface tension force $\mathbf{f}_s$ in $x$-direction. For the temporal discretization the pressure-correction method (see e.g. van Kan [57]) is employed. First a tentative velocity $u^*$ is computed by the predictor (dropping the subscript $_h$):

$$\begin{aligned} \frac{u^* - u^n}{\Delta t} = -u^n\left(\frac{\partial u}{\partial x}\right)^n - v^n\left(\frac{\partial u}{\partial y}\right)^n - w^n\left(\frac{\partial u}{\partial z}\right)^n + \\ \frac{1}{\rho^{n+\frac{1}{2}}}\left(\frac{\partial\mu\frac{\partial u}{\partial x}}{\partial x} + \frac{\partial\mu\frac{\partial u}{\partial y}}{\partial y} + \frac{\partial\mu\frac{\partial u}{\partial z}}{\partial z}\right)^* + \\ \frac{1}{\rho^{n+\frac{1}{2}}}\left(-\frac{\partial\mu\frac{\partial v}{\partial y}}{\partial x} + \frac{\partial\mu\frac{\partial v}{\partial x}}{\partial y} - \frac{\partial\mu\frac{\partial w}{\partial z}}{\partial x} + \frac{\partial\mu\frac{\partial w}{\partial x}}{\partial z}\right)^n, \end{aligned} \tag{4.30}$$

where $\mu$ is evaluated at $t^{n+\frac{1}{2}}$ due to the staggering of the Level-Set function $\Phi$, i.e.:

$$\begin{aligned} \mu^n &= \mu(\Phi^{n+\frac{1}{2}}), \\ \mu^* &= \mu(\Phi^{n+\frac{1}{2}}). \end{aligned} \tag{4.31}$$

The equations for $v$ and $w$ are treated similarly. The stress tensor is split into a part on time level $^*$ (implicit) and $^n$ (explicit), due to the fact that $u$, $v$ and

$w$ are solved sequentially. Note that all terms containing $u$ in the diffusion part of Eqn. (4.30) are implicit. In the stress tensor the continuity equation is used, such that the explicit terms in the stress tensor vanish away from the interface, where $\mu$ is constant.

The resulting system of equations is solved by an incomplete Cholesky preconditioned Conjugate Gradient (ICCG) method.

The velocities at the new time instant $^{n+1}$ are computed by:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{*}}{\Delta t} = \left(-\frac{1}{\rho}Gp + \frac{1}{\rho}\mathbf{f}_s + \mathbf{g}\right)^{n+\frac{1}{2}} \tag{4.32}$$

under the constraint of Eqn. (4.1). Note that the pressure gradient is evaluated at $t^{n+\frac{1}{2}}$. This gradient has a strong increase near the interface due to the surface tension forces $\mathbf{f}_s$. In other words, the pressure gradient is balanced by the surface tension force and the gravity vector. For that reason the term $\frac{1}{\rho}\nabla p$ is always accompanied by the surface tension force $\mathbf{f}_s$ and gravity vector $\mathbf{g}$:

$$\left(-\frac{1}{\rho}\frac{\partial p}{\partial x} + \frac{1}{\rho}f_{sx} + g_x\right)^{n+\frac{1}{2}}.$$

The pressure gradient jump and the surface tension forces move with the interface in time. When different time levels are used to evaluate pressure gradient and surface tension force, the pressure gradient is not located at the same position as the surface tension force. Furthermore, in Eqn. (4.32) $G$ is the discrete gradient operator, which comprises the Ghost-Fluid method as discussed in Section 4.4 (see Eqn. (4.23)). This gives

$$\mathbf{u}^{n+1} \quad = \quad \mathbf{u}^{*} + \Delta t \quad \left(-\frac{1}{\rho}Gp + \frac{1}{\rho}\mathbf{f}_s + \mathbf{g}\right)^{n+\frac{1}{2}} \tag{4.33}$$

and

$$D\mathbf{u}^{n+1} = 0, \tag{4.34}$$

where $D$ represents the discretization of the divergence. Combining these expressions results in:

$$D\frac{1}{\rho^{n+\frac{1}{2}}}Gp^{n+\frac{1}{2}} = D\left(\frac{1}{\Delta t}\mathbf{u}^{*} - \left(\frac{1}{\rho}\mathbf{f}_s + \mathbf{g}\right)^{n+\frac{1}{2}}\right). \tag{4.35}$$

The resulting system of equations is solved by an incomplete Cholesky preconditioned Conjugate Gradient (ICCG) method. Iterative methods might suffer from the discontinuous behavior of the coefficient $\frac{1}{\rho}$ that appears in the equations. However, the ICCG method is found to be extremely robust for the applications considered in this work.

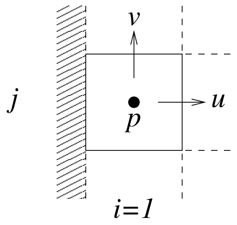


Figure 4.3: Grid layout near the boundary in two dimensions

## 4.6 Initial and Boundary conditions

As discussed in Section 2.7, the fluid is assumed to be initially at rest, i.e.

$$\mathbf{u}^0 = \mathbf{0}. \tag{4.36}$$

The Level-Set function $\Phi^{n+\frac{1}{2}}$ at $t^{\frac{1}{2}}$ is obtained by employing a first order approximation of Eqn. (3.11):

$$\Phi^{\frac{1}{2}} = \Phi^0 - (\mathbf{u}\cdot\nabla\Phi)^0\,\frac{1}{2}\Delta t = \Phi^0, \tag{4.37}$$

where the initial condition for $\mathbf{u}$ (Eqn. (4.36)) is used.

Dirichlet boundary conditions are imposed for the velocity $\mathbf{u}$. In Fig. 4.3 a two-dimensional example of the left-hand side boundary is depicted. The index for this boundary is $i = \frac{1}{2}$. Virtual values $u_{\frac{1}{2},j,k}$, $v_{-1,j+\frac{1}{2},k}$ and $w_{-1,j,k+\frac{1}{2}}$ are introduced near the boundary. The boundary conditions are then discretized as:

$$\begin{aligned} u_{\frac{1}{2},j,k} &= 0, && (4.38)\\ v_{0,j+\frac{1}{2},k} + v_{1,j+\frac{1}{2},k} &= 0, && (4.39)\\ w_{0,j,k+\frac{1}{2}} + w_{1,j,k+\frac{1}{2}} &= 0. && (4.40) \end{aligned}$$

The velocity derivatives near the boundary result in:

$$\begin{aligned} \left(\frac{\partial u}{\partial x}\right)_{1,j,k} &= \frac{u_{2,j,k}}{\Delta x}, && (4.41)\\ \left(\frac{\partial v}{\partial x}\right)_{1,j+\frac{1}{2},k} &= \frac{v_{2,j+\frac{1}{2},k}}{\Delta x}, && (4.42)\\ \left(\frac{\partial v}{\partial x}\right)_{\frac{1}{2},j+\frac{1}{2},k} &= \frac{2v_{1,j+\frac{1}{2},k}}{\Delta x}, && (4.43)\\ \left(\frac{\partial w}{\partial x}\right)_{1,j,k+\frac{1}{2}} &= \frac{w_{2,j,k+\frac{1}{2}}}{\Delta x}, && (4.44)\\ \left(\frac{\partial w}{\partial x}\right)_{\frac{1}{2},j,k+\frac{1}{2}} &= \frac{2w_{1,j,k+\frac{1}{2}}}{\Delta x}, && (4.45) \end{aligned}$$

and similar for the other boundaries.

There are no boundary conditions required for the pressure $p$, since in Eqn. (4.35) the velocity boundary conditions are included in the discrete divergence operator $D$. More details can be found in Van Kan [57].

## 4.7 Time Step criteria

Following Kang et al. [58] and Sussman et al. [44], an adaptive time stepping procedure is chosen by considering the time-step restrictions due to convection and surface tension effects. Diffusion is accounted for implicitly, hence no time-step restriction is encountered. The time-step restriction is:

$$\Delta t \leq \mathrm{CFL}\min(\Delta t_c, \Delta t_s), \tag{4.46}$$

where, following Kang et al. [58] and Sussman et al. [44], CFL $= \frac{1}{2}$ is used. Here $\Delta t_c$ is the restriction due to advection:

$$\Delta t_c = \frac{1}{\frac{|u|_{max}}{\Delta x} + \frac{|v|_{max}}{\Delta y} + \frac{|w|_{max}}{\Delta z}} \tag{4.47}$$

$\Delta t_s$ is the restriction due to surface tension. The restriction due to surface tension given by Kang et al. [58] is

$$\Delta t_s = \frac{1}{\sqrt{\frac{\sigma|\kappa|_{max}}{\min(\rho_0,\rho_1)\min(\Delta x,\Delta y)^2}}}. \tag{4.48}$$

Since the surface tension force is regularized, i.e. $\frac{1}{\rho}\sigma\kappa$ is replaced by $\frac{1}{\frac{1}{2}(\rho_0+\rho_1)}\sigma\kappa\delta(\Phi)h$ and $h = \min(\Delta x, \Delta y, \Delta z)$, the restriction becomes

$$\Delta t_s = \frac{1}{\sqrt{\frac{|\sigma\kappa\delta(\Phi)|_{max}}{\frac{1}{2}(\rho_0+\rho_1)\min(\Delta x,\Delta y,\Delta y)}}}. \tag{4.49}$$

## 4.8 Conclusion

The governing equations are discretized by a finite difference approach. The viscosity is regularized to decouple the interface conditions. For the interface forces the CSF approach is adopted. The Ghost-Fluid method for incompressible flows is used to compute derivatives near the interface. The interface advection is decoupled from the flow-field computations. The pressure correction method is used for the temporal discretization.

# Chapter 5

# Interface advection

## 5.1 Introduction

The strategy of modeling bubbly flows is to compute the flow with a given interface position and to subsequently evolve the interface in the given flow field. The flow is expressed by velocity field $\mathbf{u}$ and the interface is implicitly defined by Level-Set function $\Phi$. If symbolically the governing equations are written as (repeating Eqn. (4.24))

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} &= \mathcal{A}(\mathbf{u}, \Phi), \\ \frac{\partial \Phi}{\partial t} &= \mathcal{B}(\Phi, \mathbf{u}), \end{aligned} \tag{5.1}$$

then the decoupled temporal discretization is given by Eqns. (4.25) and (4.26):

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = A(\mathbf{u}^n, \mathbf{u}^{n+1}, \Phi^{n+\frac{1}{2}}), \tag{5.2}$$

$$\frac{\Phi^{n+\frac{3}{2}} - \Phi^{n+\frac{1}{2}}}{\Delta t} = B(\Phi^{n+\frac{1}{2}}, \Phi^{n+\frac{3}{2}}, \mathbf{u}^{n+1}), \tag{5.3}$$

where $A$ and $B$ are the discrete counterparts of operators $\mathcal{A}$ and $\mathcal{B}$ respectively and superscript $^n$ denotes time-level $n$. In the previous sections the manner by which the flow is computed with a given interface position was described. Next we consider the evolution of the interface with velocities evaluated at $t^{n+1}$.

## 5.2 Level-Set advection

The interface, say $S$, is the zero level-set of $\Phi$:

$$S(t) = \left\{ \mathbf{x} \in \mathbb{R}^3 | \Phi(\mathbf{x}, t) = 0 \right\}. \tag{5.4}$$

The interface is evolved by advecting the Level-Set function in the flow field as if it is a material property, according to Eqn. (3.11):

$$\frac{\partial \Phi}{\partial t} + \mathbf{u} \cdot \nabla \Phi = 0. \tag{5.5}$$

A homogeneous Neumann boundary condition for $\Phi$ is imposed at the boundaries. The accuracy of the approximation of Eqn. (3.11) determines the accuracy of the interface representation. This accuracy will also determine the mass errors. The discretization of the gradient of $\Phi$ can be either first order upwind, or second or third order ENO (Sussman et al. [37, 44], Chang et al. [36]). In case of first-order spatial discretization, a forward Euler temporal discretization is sufficient. In case of the higher order spatial discretization, a Runge-Kutta scheme is applied (see e.g. Sussman et al. [45]) which has the same order as the order of the spatial discretization of Eqn. (5.5).

Note that the increased order of accuracy of the Level-Set advection improves mass conservation. The global accuracy of $\Phi$ however will still depend on the accuracy of the velocity field, which is first order in this research. Higher order accuracy might not be expected when viscosity and interface forces are regularized.

## 5.3 Re-initialization

The surface tension forces and viscosity are regularized near the interface in the interval $-\alpha \leq \Phi \leq \alpha$, see Eqns. (4.5) and (4.9), where $\alpha$ is proportional to the mesh width. The regularization width is therefore the distance between the ($\Phi = -\alpha$) and ($\Phi = \alpha$) level-sets. It can be easily understood that when $|\nabla\Phi|$ becomes too large, the regularization width becomes too small and when $|\nabla\Phi|$ is too small, the regularization width will be too large. On the other hand, if $\Phi$ is a distance function, which satisfies $|\nabla\Phi| = 1$ (see Eqn. (3.14)), the regularization width will be $2\alpha$. In other words, it is necessary to control $|\nabla\Phi|$. This can be achieved by using a signed distance function for $\Phi$. But if an initial signed distance function is advected through a non-uniform flow, it does not necessarily correspond to a distance function any longer. Therefore, re-initialization as described by Sussman et al. [37] and Chang et al. [36] is applied. The Level-Set function $\Phi$ is re-initialized to make it a distance function by solving until steady state for artificial time $t'$ (see Eqn. (3.15)):

$$\begin{aligned} &\frac{\partial \Phi}{\partial t'} - \operatorname{sign}(\Phi^0)\,(1 - \|\nabla\Phi\|) = 0, \\ &\Phi|_{t'=0} = \Phi^0, \end{aligned} \tag{5.6}$$

where the initial condition $\Phi^0$ results from the Level-Set advection, Eqn. (5.5). A homogeneous Neumann boundary condition for $\Phi$ is used. The spatial discretization of Eqn. (5.6) is (see e.g. Sethian [41]):

$$\frac{\mathrm{d}\Phi_{i,j,k}}{\mathrm{d}t'} = \begin{cases} 1 - \sqrt{\begin{array}{l} \max\left({d_{x,L}^{+}}^{2}, {d_{x,R}^{-}}^{2}\right) + \\ \max\left({d_{y,L}^{+}}^{2}, {d_{y,R}^{-}}^{2}\right) + \\ \max\left({d_{z,L}^{+}}^{2}, {d_{z,R}^{-}}^{2}\right) \end{array}} & \Phi^0 > 0 \\ \\ -1 + \sqrt{\begin{array}{l} \max\left({d_{x,L}^{-}}^{2}, {d_{x,R}^{+}}^{2}\right) + \\ \max\left({d_{y,L}^{-}}^{2}, {d_{y,R}^{+}}^{2}\right) + \\ \max\left({d_{z,L}^{-}}^{2}, {d_{z,R}^{+}}^{2}\right) \end{array}} & \Phi^0 < 0 \\ 0 & \Phi^0 = 0, \end{cases} \tag{5.7}$$

where $d_{x,L/R}$ are the differences that are approximated by an ENO scheme

$$\begin{array}{rcl} d_{x,L} & = & \left(\frac{\partial\Phi}{\partial x}\right)_{i,j,k\,L}, \\ d_{x,R} & = & \left(\frac{\partial\Phi}{\partial x}\right)_{i,j,k\,R}. \end{array} \tag{5.8}$$

In this expression $L$ and $R$ indicate the starting point in the ENO scheme. For example, a first-order approximation are the one-sided differentials:

$$\begin{array}{rcl} d_{x,L} & = & \frac{\Phi_{i,j,k} - \Phi_{i-1,j,k}}{\Delta x}, \\ d_{x,R} & = & \frac{\Phi_{i+1,j,k} - \Phi_{i,j,k}}{\Delta x}, \end{array} \tag{5.9}$$

and analogously for $d_{y,L}$, $d_{y,R}$, $d_{z,L}$ and $d_{z,R}$. Furthermore, $^{+}$ is the positive part and $^{-}$ the negative part:

$$\begin{array}{l} d^{+} = \max(d, 0) \\ d^{-} = \min(d, 0). \end{array} \tag{5.10}$$

Time integration is performed by means of a Runge-Kutta method that has the same order as the error of the spatial discretization, which in turn matches the order of the discretization error of the Level-Set advection equation, Eqn. (5.5). For a stable time integration of Eqn. (5.6) a constraint is necessary on $\Delta t'$. To find such a constraint Eqn. (5.6) is rewritten as

$$\frac{\partial\Phi}{\partial t'} + \mathbf{q} \cdot \nabla\Phi = \mathrm{sign}(\Phi^0), \tag{5.11}$$

which is a convection equation for $\Phi$ with convection velocity $\mathbf{q} = (g_x, g_y, q_z)^t$:

$$\mathbf{q} = \mathrm{sign}(\Phi^0)\frac{\nabla\Phi}{\|\nabla\Phi\|}. \tag{5.12}$$

Using $|q_x| < 1$, $|q_y| < 1$ and $|q_z| < 1$ in the CFL condition leads to the following time-step:

$$\Delta t' = \frac{\sigma_{\mathrm{reinit}}}{\frac{1}{\Delta x} + \frac{1}{\Delta y} + \frac{1}{\Delta z}}, \tag{5.13}$$

where $\sigma_{\text{reinit}} \leq 1$ is the Courant number. In this research $\sigma_{\text{reinit}} = \frac{1}{2}$ is used.

The difficulty with re-initialization is that the interface position is only defined in the pseudo-initial condition $\Phi^0$. The consequence is that the interface shifts during the pseudo-time stepping due to discretization errors and hence mass is lost. The re-initialization procedure has therefore been modified as discussed in Chapter 6.

## 5.4 Mass-Conserving Level-Set advection

For the reasons given in Chapter 3 this work is based on a Level-Set method. The major disadvantage of the Level-Set method is that it is not mass-conserving. Additional effort is necessary and in this research the Volume-of-Fluid function $\Psi$ is used. This is performed without applying the difficult convection step (namely interface reconstruction) which makes VOF so elaborate.

Since density is piecewise constant, volume conservation yields mass conservation. The volume $V_k$ of a certain fluid within a computational cell $\Omega_k$ is

$$V_k = \int\limits_{\Omega_k} \chi \, \mathrm{d}\Omega, \tag{5.14}$$

where $\chi$ is a color function that is 1 in fluid '1' and '0' elsewhere, see Eqn. (2.3) and Fig. 2.1. This can be expressed in terms of the Level-Set function $\Phi$ as:

$$V_k = \int\limits_{\Omega_k} H(\Phi) \, \mathrm{d}\Omega, \tag{5.15}$$

where $H$ is the Heaviside step function. Although $\Phi$ might be conserved,i.e.

$$\int\limits_{\Omega} \Phi^{n+\frac{3}{2}} \, \mathrm{d}\Omega = \int\limits_{\Omega} \Phi^{n+\frac{1}{2}} \, \mathrm{d}\Omega, \tag{5.16}$$

this does not imply that the sign of $\Phi$ (hence volume) is conserved:

$$\int\limits_{\Omega} H(\Phi^{n+\frac{3}{2}}) \, \mathrm{d}\Omega \neq \int\limits_{\Omega} H(\Phi^{n+\frac{1}{2}}) \, \mathrm{d}\Omega, \tag{5.17}$$

where $\Omega$ is the whole computational domain. Consequently volume, and hence mass, are not conserved:

$$\sum_k V_k^{n+\frac{3}{2}} \neq \sum_k V_k^{n+\frac{1}{2}}, \tag{5.18}$$

where the summation is over all computational cells $k$. The volume errors can be made smaller by increasing the accuracy of the discretization. For this reason higher-order ENO schemes and Runge-Kutta methods are often applied. In this research we use first-order Level-Set advection and re-initialization. Low order advection and re-initialization ensure numerical smoothness of $\Phi$. Furthermore, when the flow-field is computed, higher order accuracy might not be expected when the CSF method is applied and viscosity is regularized. In that respect, higher order discretization of Eqn. (3.11) will only lead to improved mass conservation for the pure Level-Set methods.

The Volume-of-Fluid method (see Section 3.4.1) on the other hand conserves mass by construction. With Eqn. (3.3) it is clear that

$$V_k = \Psi_k \mathrm{vol}(\Omega_k), \tag{5.19}$$

where $\Psi$ is the Volume-of-Fluid function that measures the fractional volume of fluid '1' within a computational cell. Rewriting Eqn. (3.8) as

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathrm{vol}(\Omega_k)\Psi(\mathbf{x}_k)\right) + \int\limits_{\partial\Omega_k} \chi\mathbf{u}\cdot\mathbf{n}\,\mathrm{d}S = 0 \tag{5.20}$$

shows that volume, and hence mass, is conserved if $\Psi$ is advected conservatively, i.e.

$$\sum_k \Psi_k^{n+\frac{3}{2}}\mathrm{vol}(\Omega_k) = \sum_k \Psi_k^{n+\frac{1}{2}}\mathrm{vol}(\Omega_k). \tag{5.21}$$

In order to conserve mass with the Level-Set method, corrections to the Level-Set function are made by considering the fractional volume $\Psi$ of a certain fluid within a computational cell. First the usual Level-Set advection is performed: first-order advection and re-initialization as described above. Since the obtained Level-Set function $\Phi^*$ will certainly not conserve mass, corrections to $\Phi^*$ are made such that mass is conserved. This requires three steps:

1. the relative volume of a certain fluid in a computational cell (called 'volume-of-fluid' function $\Psi$) is to be computed from the Level-Set function $\Phi^{n+\frac{1}{2}}$: $\Psi^{n+\frac{1}{2}} = f(\Phi^{n+\frac{1}{2}}, \nabla\Phi^{n+\frac{1}{2}})$;
2. the volume-of-fluid function has to be advected conservatively during a time step towards $\Psi^{n+\frac{3}{2}}$;
3. with this new volume-of-fluid function $\Psi^{n+\frac{3}{2}}$, corrections to $\Phi^*$ are sought such that $f(\Phi^{n+\frac{3}{2}}, \nabla\Phi^{n+\frac{3}{2}}) = \Psi^{n+\frac{3}{2}}$ holds.

These three steps will now be described in more detail.

### 5.4.1 Step 1: Volume-of-Fluid function

In the computational domain two functions $\Phi : \mathrm{I\!R}^3 \to \mathrm{I\!R}$ and $\Psi : \mathrm{I\!R}^3 \to \mathrm{I\!R}$ are considered. These are the Level-Set function $\Phi(\mathbf{x})$ and the Volume-of-Fluid function $\Psi(\mathbf{x}_k)$ respectively, where $\mathbf{x} \in \mathrm{I\!R}^3$ and $k$ is the index of the computational cell. The Volume-of-Fluid function $\Psi$ in a computational cell $\Omega_k$ is defined by Eqn. (3.3), which is in terms of the Level-Set methodology:

$$\Psi(\mathbf{x}_k) = \frac{1}{\mathrm{vol}(\Omega_k)} \int\limits_{\Omega_k} H(\Phi)\,\mathrm{d}\Omega, \tag{5.22}$$

where $H$ is the Heaviside step function and $\Omega_k$ is a cube with sizes $\Delta x$, $\Delta y$ and $\Delta z$. Assume that $\mathbf{x}_k$ corresponds to the center of the cube $\Omega_k$. The linearization of $\Phi$ around $\mathbf{x}_k$ is called $\varphi$:

$$\varphi(\mathbf{y};\Phi,\nabla\Phi,\mathbf{x}_k) = \Phi_k + (\tfrac{\partial\Phi}{\partial x},\tfrac{\partial\Phi}{\partial y},\tfrac{\partial\Phi}{\partial z})_k^t \cdot (\mathbf{y}-\mathbf{x}_k), \tag{5.23}$$

where the derivatives are approximated by central differences. The linearization of $\Phi$ can be used to approximate $\Psi$:

$$\Psi(\mathbf{x}_k) \approx f(\Phi_k, \nabla\Phi_k). \tag{5.24}$$

The goal is to find $f$. To ease the analysis considerably, map $\Omega$ onto a unit cube with coordinates $(\xi,\eta,\zeta) \in (-\frac{1}{2},\frac{1}{2})^3$. The linearization can be written as

$$\varphi = \Phi_k + D_\xi\xi + D_\eta\eta + D_\zeta\zeta, \tag{5.25}$$

where the axes are chosen such that

$$D_\xi \geq D_\eta \geq D_\zeta \geq 0. \tag{5.26}$$

This choice will limit the number of possible interface topologies in the cube. As a result

$$\begin{aligned}
D_\xi &= \max(|\Delta x\big(\tfrac{\partial\Phi}{\partial x}\big)_k|, |\Delta y\Big(\tfrac{\partial\Phi}{\partial y}\Big)_k|, |\Delta z\big(\tfrac{\partial\Phi}{\partial z}\big)_k|),\\
D_\zeta &= \min(|\Delta x\big(\tfrac{\partial\Phi}{\partial x}\big)_k|, |\Delta y\Big(\tfrac{\partial\Phi}{\partial y}\Big)_k|, |\Delta z\big(\tfrac{\partial\Phi}{\partial z}\big)_k|),\\
D_\eta &= |\Delta x\big(\tfrac{\partial\Phi}{\partial x}\big)_k| + |\Delta y\Big(\tfrac{\partial\Phi}{\partial y}\Big)_k| + |\Delta z\big(\tfrac{\partial\Phi}{\partial z}\big)_k| - D_\xi - D_\zeta.
\end{aligned} \tag{5.27}$$

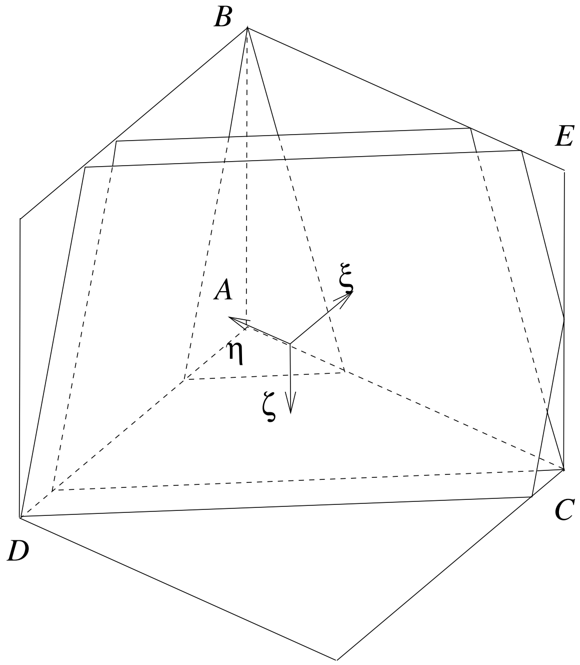
Since $\Phi$ is linearized, the zero level-set of $\Phi$ (the interface) is a plane.

The Volume-of-Fluid function will be derived geometrically by computing the relative volume enclosed by the cut plane within $\Omega_k$. It is important to take into account the topology of the cut plane. The topology of the cut plane changes when it passes a corner of the cube, see Fig. 5.1. Due to symmetry, only the cases $\Phi_k \leq 0$ will be considered here. Two cases are defined, see Fig. 5.1:

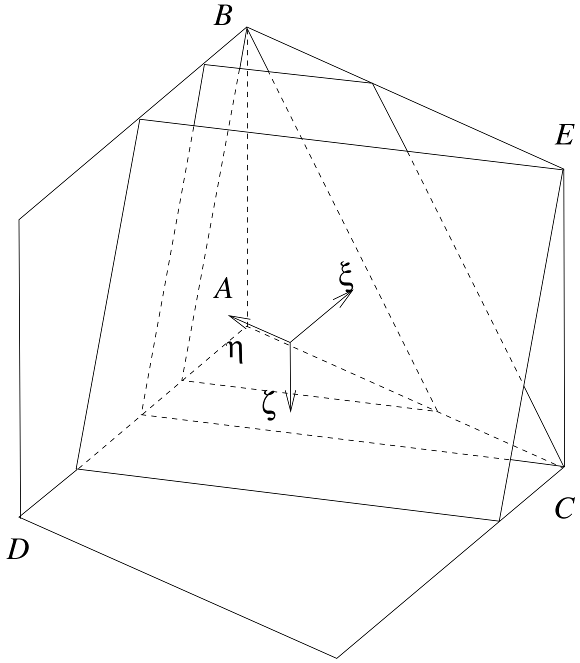
$$\begin{array}{lcl} \text{case I} & : & D_\xi \leq D_\eta + D_\zeta \\ \text{case II} & : & D_\xi > D_\eta + D_\zeta \end{array} \tag{5.28}$$

Note that there are only two cases due to the choice of $\xi$, $\eta$ and $\zeta$. In Fig. 5.2 a geometrical representation is shown which can be used to compute $f$. This is achieved by using the pyramids at corners $A$, $B$, etcetera. These have length, width and height $\frac{\Phi_A}{D_\xi}$, $\frac{\Phi_A}{D_\eta}$, $\frac{\Phi_A}{D_\zeta}$ respectively and similar for the other corners. Here $\Phi_A$ is the value $\varphi(\mathbf{x}_A)$ at corner $A$, etcetera:

$$\begin{aligned}
\Phi_A &= \Phi_k + \tfrac{1}{2}D_\xi + \tfrac{1}{2}D_\eta + \tfrac{1}{2}D_\zeta\\
\Phi_B &= \Phi_k + \tfrac{1}{2}D_\xi + \tfrac{1}{2}D_\eta - \tfrac{1}{2}D_\zeta\\
\Phi_C &= \Phi_k + \tfrac{1}{2}D_\xi - \tfrac{1}{2}D_\eta + \tfrac{1}{2}D_\zeta\\
\Phi_D &= \Phi_k - \tfrac{1}{2}D_\xi + \tfrac{1}{2}D_\eta + \tfrac{1}{2}D_\zeta\\
\Phi_E &= \Phi_k + \tfrac{1}{2}D_\xi - \tfrac{1}{2}D_\eta - \tfrac{1}{2}D_\zeta.
\end{aligned} \tag{5.29}$$

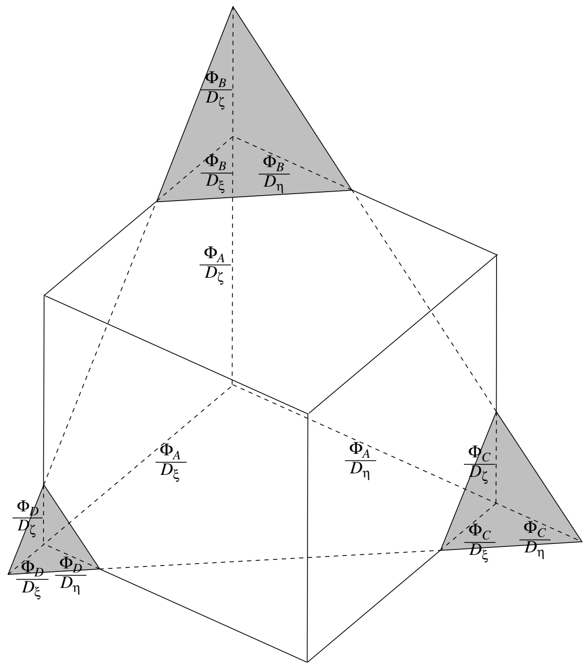


(a) case I: $D_\xi \leq D_\eta + D_\zeta$

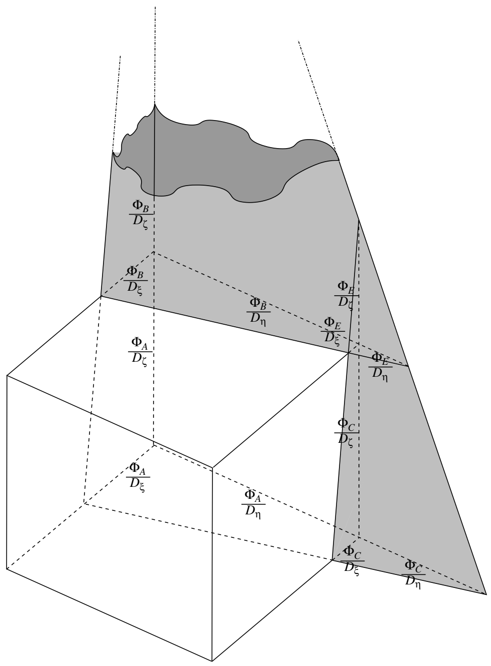


(b) case II: $D_\xi > D_\eta + D_\zeta$

Figure 5.1: Topologies of the cut plane for $\Phi \leq 0$



(a) case I



(b) case II

Figure 5.2: Volumes for $\Phi \leq 0$

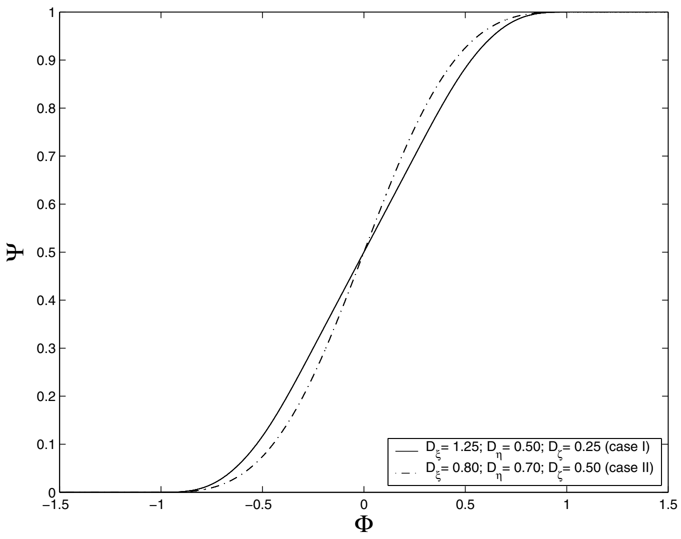


Figure 5.3: Examples of Volume-of-Fluid function for the two cases

Function $f$ then follows as:

$$f = \frac{\mathcal{A}}{6 D_\xi D_\eta D_\zeta} \qquad \Phi \le 0 \tag{5.30}$$

and

$$f = 1 - f(-\Phi, \nabla\Phi) \qquad \Phi > 0, \tag{5.31}$$

where

$$\begin{aligned} \mathcal{A} \;=\; & \max(\Phi_A, 0)^3 - \max(\Phi_B, 0)^3 - \\ & \max(\Phi_C, 0)^3 - \max(\Phi_D, 0)^3 + \\ & \max(\Phi_E, 0)^3. \end{aligned} \tag{5.32}$$

Examples for the cases in (5.28) are shown in Fig. 5.3.

Numerical difficulties arise when the denominator vanishes in Eqn. (5.30). Because of Eqn. (5.26), this occurs when

$$\begin{cases} D_\zeta = 0, \text{or} \\ D_\zeta = 0 \wedge D_\eta = 0, \text{or} \\ D_\zeta = 0 \wedge D_\eta = 0 \wedge D_\xi = 0, \end{cases} \tag{5.33}$$

which are the two-, one- and zero-dimensional limits of the problem respectively. These limits will be taken in that order. First note that because of inequality (5.26)

$$-\frac{1}{2} D_\xi + \frac{1}{2} D_\eta \le 0, \tag{5.34}$$

which yields

$$\Phi_D \leq \Phi + \frac{1}{2}D_\zeta. \tag{5.35}$$

Since $\Phi \leq 0$, the following upper bound is found for $\Phi_D$

$$\Phi_D \leq \frac{1}{2}D_\zeta, \tag{5.36}$$

so that

$$\lim_{D_\zeta \to 0} \frac{\max(\Phi_D, 0)^3}{D_\zeta} = 0. \tag{5.37}$$

Furthermore, for $|\Phi + \frac{1}{2}D_\xi + \frac{1}{2}D_\eta| > \frac{1}{2}D_\zeta$, $\Phi_A$ and $\Phi_B$ have the same sign, so that

$$\begin{aligned} \lim_{D_\zeta \to 0} \tfrac{\max(\Phi_A,0)^3 - \max(\Phi_B,0)^3}{D_\zeta} &= 3\max(\Phi + \tfrac{1}{2}D_\xi + \tfrac{1}{2}D_\eta, 0)^2 \\ &= 3\max(\left.\Phi_A\right|_{D_\zeta=0}, 0)^2. \end{aligned} \tag{5.38}$$

Otherwise $0 \leq \Phi_A \leq D_\zeta$ and $\Phi_B \leq 0$ so that

$$\lim_{D_\zeta \to 0} \tfrac{\max(\Phi_A,0)^3 - \max(\Phi_B,0)^3}{D_\zeta} = 0. \tag{5.39}$$

If $|\Phi + \frac{1}{2}D_\xi - \frac{1}{2}D_\eta| > \frac{1}{2}D_\zeta$, then $\Phi_C$ and $\Phi_E$ have the same sign so that

$$\begin{aligned} \lim_{D_\zeta \to 0} \tfrac{\max(\Phi_C,0)^3 - \max(\Phi_E,0)^3}{D_\zeta} &= 3\max(\Phi + \tfrac{1}{2}D_\xi - \tfrac{1}{2}D_\eta, 0)^2 \\ &= 3\max(\left.\Phi_C\right|_{D_\zeta=0}, 0)^2. \end{aligned} \tag{5.40}$$

Otherwise $0 \leq \Phi_C \leq D_\zeta$ and $\Phi_E \leq 0$ so that

$$\lim_{D_\zeta \to 0} \tfrac{\max(\Phi_C,0)^3 - \max(\Phi_E,0)^3}{D_\zeta} = 0. \tag{5.41}$$

Combining these results, the two-dimensional limit follows as

$$\lim_{D_\zeta \to 0} f = \frac{\max(\left.\Phi_A\right|_{D_\zeta=0}, 0)^2 - \max(\left.\Phi_C\right|_{D_\zeta=0}, 0)^2}{2D_\xi D_\eta} \qquad \Phi \leq 0. \tag{5.42}$$

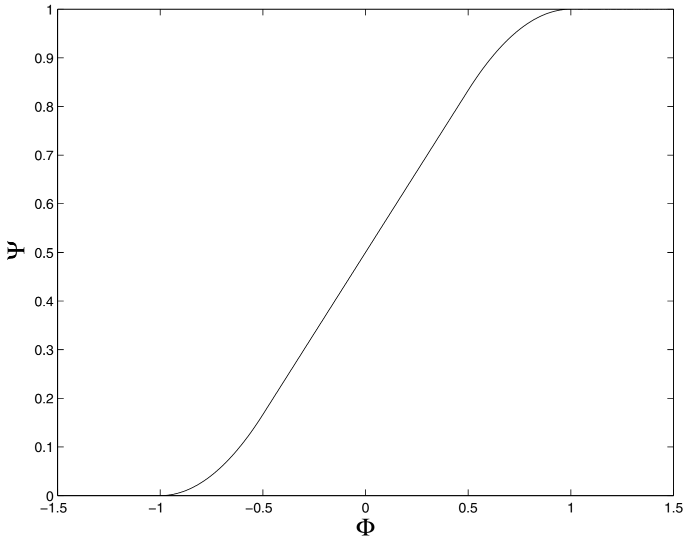
An example is shown in Fig. 5.4(a). The one-dimensional limit follows in a similar fashion from this equation:

$$\lim_{D_\eta \to 0} \lim_{D_\zeta \to 0} f = \frac{\max(\left.\Phi_A\right|_{D_\eta=0, D_\zeta=0}, 0)}{D_\xi} \qquad \Phi \leq 0. \tag{5.43}$$

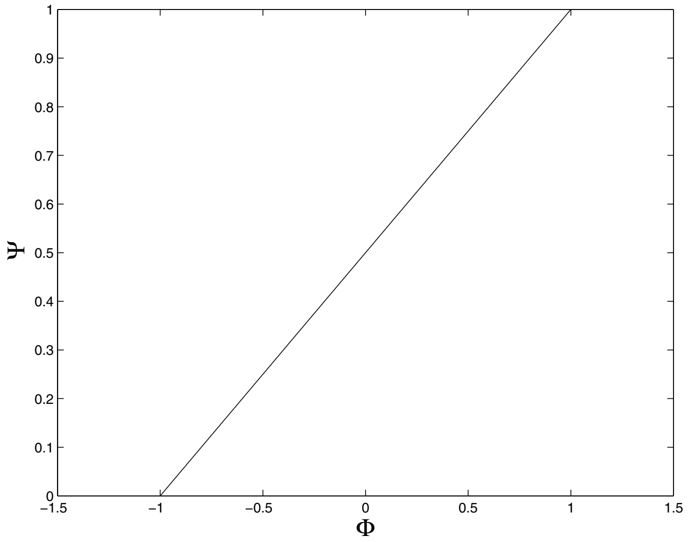
An example is shown in Fig. 5.4(b). Finally

$$\lim_{D_\xi \to 0} \lim_{D_\eta \to 0} \lim_{D_\zeta \to 0} f = \begin{cases} 0 & \Phi < 0 \\ \frac{1}{2} & \Phi = 0 \\ 1 & \Phi > 0, \end{cases} \tag{5.44}$$

which is a step function.



(a) $D_\xi = 0$; $D_\eta = 1.5$; $D_\zeta = 0.5$



(b) $D_\xi = 0$; $D_\eta = 0$; $D_\zeta = 2$

Figure 5.4: Examples of the Volume-of-Fluid function

### 5.4.2 Step 2: Volume-of-Fluid advection

The evolution of the Volume-of-Fluid function $\Psi$ is given by Eqn. (3.8), which in terms of the Level-Set methodology is:

$$\frac{\mathrm{d}\Psi(\mathbf{x}_k)}{\mathrm{d}t} + \frac{1}{\mathrm{vol}(\Omega_k)} \int\limits_{\partial\Omega_k} H(\Phi)\mathbf{u}\cdot\mathbf{n}\,\mathrm{d}S = 0. \tag{5.45}$$

The Volume-of-Fluid function after a time step is found by considering the flux of fluid that flows through the boundaries of a computational cell during time-step $\Delta t$. Consider a face $\Gamma$ of the control volume $\Omega_k$. The flux $F$ through the boundary $\Gamma$ is

$$F = \int\limits_{\Delta t} \int\limits_{\partial\Omega_k} H(\Phi(\mathbf{x}, t+\tau))\ \mathbf{u}(\mathbf{x}, t+\tau)\cdot\mathbf{n}\ \ \mathrm{d}S\,\mathrm{d}\tau. \tag{5.46}$$

This flux can be rewritten as

$$F = \int\limits_{\Omega_D} H(\Phi(\mathbf{x}, t))\,\mathrm{d}\Omega, \tag{5.47}$$

where $\Omega_D$ is the *donating region* of face $\Gamma$, which initially contains all fluid that will flow through face $\Gamma$ during time-step $\Delta t$ (see Fig. 5.5). Summation over all boundary faces leads for computational cell $(i, j, k)$ to:

$$\Psi^{n+\frac{3}{2}}_{i,j,k} = \Psi^{n+\frac{1}{2}}_{i,j,k} - \frac{1}{\Delta x\Delta y\Delta z} \Big( \begin{array}{lcl} F_{x\,i+\frac{1}{2},j,k} & - & F_{x\,i-\frac{1}{2},j,k} + \\ F_{y\,i,j+\frac{1}{2},k} & - & F_{y\,i,j-\frac{1}{2},k} + \\ F_{z\,i,j,k+\frac{1}{2}} & - & F_{z\,i,j,k-\frac{1}{2}} \Big), \end{array} \tag{5.48}$$

where the subscripts indicate the corresponding boundary face. Depending on the sign of the velocity at the face, the donating region can either be on the left-hand or at the right-hand side neighboring cell. Formally, the flux can therefore be split into a contribution from both neighbors, called $F^+$ and $F^-$ respectively (see Fig. 5.5 for the two-dimensional case). Of course if $F^+_{\dots} \neq 0$ then $F^-_{\dots} = 0$ and vice-versa. In this way the fluxes in $x$, $y$ and $z$ direction at a face can be written as (omitting the subscripts):

$$F_x = F^+_x + F^-_x, \qquad F_y = F^+_y + F^-_y, \qquad F_z = F^+_z + F^-_z. \tag{5.49}$$

Since the fluxes measure the fractional volume in the denoting region, Eqn. (5.24) can be applied with some scaling of variables. For the fluxes in $z$ direction for example:

$$\begin{array}{rcl} F^+_{z\ i,j,k+\frac{1}{2}} & = & \Delta x\Delta y\Delta z\nu^+\ f\left(\widehat{\Phi}_L, (\partial_x\Phi_L, \partial_y\Phi_L, \nu^+\partial_z\Phi_L)^t\right) \\ F^-_{z\ i,j,k+\frac{1}{2}} & = & \Delta x\Delta y\Delta z\nu^-\ f\left(\widehat{\Phi}_R, (\partial_x\Phi_R, \partial_y\Phi_R, -\nu^-\partial_z\Phi_R)^t\right). \end{array} \tag{5.50}$$

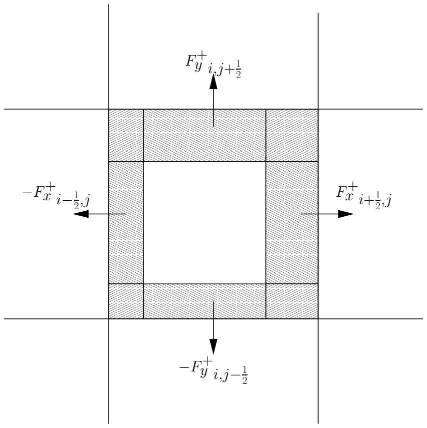

$F^{+}_{y\,i,j+\frac{1}{2}}$
$-F^{+}_{x\,i-\frac{1}{2},j}$
$F^{+}_{x\,i+\frac{1}{2},j}$
$-F^{+}_{y\,i,j-\frac{1}{2}}$

Figure 5.5: Donating regions for fluxes $F_x$ and $F_y$. These are the shaded areas that will flow through the boundaries during a time step. Doubly fluxed areas exist near the corners of the cell.

where

$$\nu^{+} = \frac{\max(u,0)\Delta t}{\Delta x}, \quad \nu^{-} = \frac{\min(u,0)\Delta t}{\Delta x}, \tag{5.51}$$

and

$$\begin{array}{rclrcl} \Phi_L &=& \Phi_{i,j,k}, & \Phi_R &=& \Phi_{i,j,k+1}, \\ \partial_x \Phi_L &=& \left.\frac{\partial \Phi}{\partial x}\right|_{i,j,k}, & \partial_x \Phi_R &=& \left.\frac{\partial \Phi}{\partial x}\right|_{i,j,k+1}, \\ \partial_y \Phi_L &=& \left.\frac{\partial \Phi}{\partial y}\right|_{i,j,k}, & \partial_y \Phi_R &=& \left.\frac{\partial \Phi}{\partial y}\right|_{i,j,k+1}, \\ \partial_z \Phi_L &=& \left.\frac{\partial \Phi}{\partial z}\right|_{i,j,k}, & \partial_z \Phi_R &=& \left.\frac{\partial \Phi}{\partial z}\right|_{i,j,k+1}. \end{array} \tag{5.52}$$

and

$$\widehat{\Phi}_L = \Phi_L + \tfrac{1}{2}(1-\nu^{+})\Delta z \partial_z \Phi_L, \quad \widehat{\Phi}_R = \Phi_R - \tfrac{1}{2}(1+\nu^{-})\Delta z \partial_z \Phi_R. \tag{5.53}$$

Time-step $\Delta t$ and velocity $u$ are included in $\nu^{+}$ and $\nu^{-}$. This scaling of variables makes the advection of $\Psi$ rather straightforward, since an analytic expression for the function $f$ is given in Eqns. (5.30) and (5.31). The fluxes in the other direction are obtained in the same way.

Fig. 5.5 illustrates that overlapping donating regions can exist in the corners of the cell. Fluid in those overlapping regions is fluxed more than once through different faces. This can be remedied by employing either a multidimensional scheme or flux-splitting. For simplicity we have chosen the second approach. The order of fluxing is: first in $x$-direction, then in $y$-direction and then in

$z$-direction. The flux-splitting of Sussman and Puckett [13] is adopted:

$$\begin{aligned}
\Psi^{(x)}_{i,j,k} &= \frac{\Psi^{n+\frac{1}{2}}_{i,j,k} - \frac{1}{\Delta x \Delta y \Delta z}\left( {F_x}^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k} - {F_x}^{n+\frac{1}{2}}_{i-\frac{1}{2},j,k} \right)}{1 - \frac{\Delta t}{\Delta x}(u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k})}, \\
\Psi^{(y)}_{i,j,k} &= \frac{\Psi^{(x)}_{i,j,k} - \frac{1}{\Delta x \Delta y \Delta z}\left( {F_y}^{(x)}_{i,j+\frac{1}{2},k} - {F_y}^{(x)}_{i,j-\frac{1}{2},k} \right)}{1 - \frac{\Delta t}{\Delta y}(v_{i,j+\frac{1}{2},k} - v_{i,j-\frac{1}{2},k})}, \\
\Psi^{(z)}_{i,j,k} &= \frac{\Psi^{(y)}_{i,j,k} - \frac{1}{\Delta x \Delta y \Delta z}\left( {F_z}^{(y)}_{i,j,k+\frac{1}{2}} - {F_z}^{(y)}_{i,j,k-\frac{1}{2}} \right)}{1 - \frac{\Delta t}{\Delta z}(w_{i,j,k+\frac{1}{2}} - w_{i,j,k-\frac{1}{2}})}, \qquad (5.54) \\
\Psi^{n+\frac{3}{2}}_{i,j,k} &= \Psi^{(z)}_{i,j,k} - \Delta t \Big( \Psi^{(x)}_{i,j,k} \frac{u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}}{\Delta x} + \\
&\qquad \Psi^{(y)}_{i,j,k} \frac{v_{i,j+\frac{1}{2},k} - v_{i,j-\frac{1}{2},k}}{\Delta y} + \\
&\qquad \Psi^{(z)}_{i,j,k} \frac{w_{i,j,k+\frac{1}{2}} - w_{i,j,k-\frac{1}{2}}}{\Delta z} \Big).
\end{aligned}$$

Due to the decoupling of interface advection and flow computation (see Eqn. (5.3)), the velocities in Eqns. (5.51) and (5.54) are evaluated at $t^{n+1}$. The fluxes ${F_y}^{(x)}$ are computed with a corrected Level-Set function $\Phi^{(x)} : f(\Phi^{(x)}, \nabla\Phi^{(x)}) = \Psi^{(x)}$ and similar for ${F_z}^{(y)}$. Note that any other flux (or operator)-splitting technique could be adopted. Note also that due to the construction of Eqn. (5.48) the quantity $\Psi$ is conserved, which is necessary for mass conservation.

### 5.4.3 Step 3: Inverse function

The Volume-of-Fluid function $\Psi^{n+\frac{3}{2}}$ was computed in the previous step. The Level-Set function after Level-Set advection as described in Section 5.2 is called $\Phi^*$. Note that $\Phi^*$ will in general not conserve mass, i.e.

$$f(\Phi^*_k, \nabla\Phi^*_k) \neq \Psi^{n+\frac{1}{2}}_k. \qquad (5.55)$$

In order to conserve mass within each computational cell, a correction $\Delta\Phi$ to $\Phi^*$ is sought, such that mass is conserved within each computational cell:

$$\left| f(\Phi^{n+\frac{3}{2}}_k, \nabla\Phi^{n+\frac{3}{2}}_k) - \Psi^{n+\frac{3}{2}}_k \right| \leq \epsilon \quad \forall k, \qquad (5.56)$$

where $\Phi^{n+\frac{3}{2}} = \Phi^* + \Delta\Phi$ is the Level-Set function at the new time-level, $f$ is the Volume-of-Fluid function (step 1) and $\epsilon$ is some tolerance. It will be clear that due to the behavior of $\Psi$ no unique solution $\Phi$ exists. On the other hand, it can be expected that the corrections $\Delta\Phi$ are small, since the mass errors in $\Phi^*$ are due to the truncation error of the discretization of Eqn. (3.11). Therefore a simple iterative approach is taken, namely by means of Picard iterations. Keeping the gradients of $\Phi$ fixed from the previous iteration, a new update for $\Phi$ is found by computing

$$\Phi^{(l+1)}_k(\mathbf{x}) = \begin{cases} g(\Psi^{n+\frac{3}{2}}_k, \nabla\Phi^{(l)}_k) & |\Psi^{(l)}_k - \Psi^{n+\frac{3}{2}}_k| > \epsilon \\ \Phi^{(l)}_k & |\Psi^{(l)}_k - \Psi^{n+\frac{3}{2}}_k| \leq \epsilon \end{cases} \quad \forall k, \qquad (5.57)$$

where $\Psi_k^{(l)} = f(\Phi_k^{(l)}, \nabla\Phi_k^{(l)})$, $\epsilon$ is a tolerance, typically $\epsilon = 10^{-8}$ and $g$ is the inverse function of $f$ with respect to its first argument:

$$f(g(\Psi, \nabla\Phi), \nabla\Phi) = \Psi. \tag{5.58}$$

The iterations are stopped if $|\Psi_k^{n+\frac{3}{2}} - \Psi_k^{(l+1)}| \leq \epsilon$, $\forall k$.

An analytical expression for $g$ in two dimensions is presented in [60]. However, expressing $g$ analytically in three dimensions is a complicated task. Instead Newton iterations are used to compute $\Phi_k^{(l+1)}$:

$$\Phi_k^{(l+1),(m+1)} = \Phi_k^{(l+1),(m)} + \frac{\Psi_k^{n+\frac{3}{2}} - f(\Phi_k^{(l),(m)}, \nabla\Phi_k^{(l)})}{\frac{\partial f}{\partial \Phi}(\Phi_k^{(l),(m)}, \nabla\Phi_k^{(l)})}. \tag{5.59}$$

The derivative $\frac{\partial f}{\partial \Phi}$ is computed analytically by straightforward differentiation. The iterations are stopped if $|\Psi_k^{n+\frac{3}{2}} - f(\dots)| \leq \epsilon$. A graphical overview of the method is given in Fig. 5.6.

### 5.4.4 Mass redistribution

For the advection of the Volume-of-Fluid function $\Psi$ (step 2, Section 5.4.2) operator splitting is applied. However, undershoots and/or overshoots can still occur as reported by Sussman et al.[13]. This leads to unphysical values of $\Psi$, namely $< 0$ and $> 1$. This is due to the non-vanishing source term in Eqn. (5.54):

$$\Delta t \left( \Psi_{i,j,k}^{(x)} \left( \frac{\partial u}{\partial x} \right)_{i,j,k} + \Psi_{i,j,k}^{(y)} \left( \frac{\partial v}{\partial y} \right)_{i,j,k} + \Psi_{i,j,k}^{(z)} \left( \frac{\partial w}{\partial z} \right)_{i,j,k} \right).$$

If the unphysical values are replaced by 0 or 1, mass errors arise which are in general of the order $10^{-4}$ (Sussman and Puckett [13]). This is also observed with the present method. Redistribution of $\Psi$ avoids these mass errors. The idea is to flux mass out of cells with $\Psi > 1$ and flux mass into cells with $\Psi < 0$.

Besides the unphysical values of $\Psi$, mass can also be distributed without the presence of an interface. This is the case when the Volume-of-Fluid function $\Psi$ is between and not equal to 0 and 1, but no interface is present within the corresponding computational cell. In that case a feature exists with a size smaller than a computational cell. Although formally not a mass error, it is not taken into account by the Ghost-Fluid method when the flow-field is computed, and will be called 'numerical vapor'. However, if numerical vapor would be bluntly removed, it would cause mass errors.

There will not be any interface in a control volume $\Omega_{i,j,k}$ around $x_{i,j,k}$ if no interface is present between $x_{i,j,k}$ and $x_{i\pm1,j,k}$ and between $x_{i,jk}$, and $x_{i,j\pm1,k}$ and between $x_{i,j,k}$ and $x_{i,j,k\pm1}$. No interface is present between two nodes if the Level-Set function $\Phi$ does not change sign. Consequently, no interface will be present in the control volume $\Omega_{i,j,k}$ if $\Phi_{i,j,k}\Phi_{i\pm1,j,k} > 0$ and $\Phi_{i,j,k}\Phi_{i,j\pm1,k} > 0$ and $\Phi_{i,j,k}\Phi_{i,j,k\pm1} > 0$.

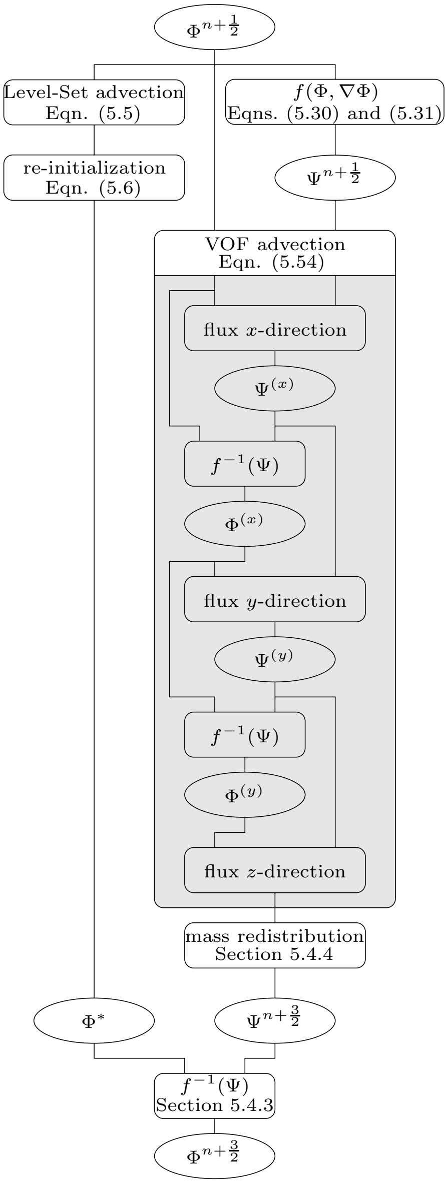


Figure 5.6: MCLS method: interface advection; $\Phi$: Level-Set function; $\Psi$: Volume-of-Fluid function. The left-hand side branch corresponds to pure Level-Set advection. The right-hand side branch represents the Volume-of-Fluid advection.

Summarizing, mass errors arise when

$$\begin{array}{lllll} \text{operator splitting:} & \Psi_{i,j,k} < 0 \quad \vee \quad \Psi_{i,j,k} > 1 & & & \\ \text{numerical vapor:} & \Psi_{i,j,k} \in (0,1) \quad \wedge & & & \\ & \Phi_{i,j,k}\Phi_{i+1,j,k} > 0 & \wedge & \Phi_{i,j,k}\Phi_{i-1,j,k} > 0 & \wedge \\ & \Phi_{i,j,k}\Phi_{i,j+1,k} > 0 & \wedge & \Phi_{i,j,k}\Phi_{i,j-1,k} > 0 & \wedge \\ & \Phi_{i,j,k}\Phi_{i,j,k+1} > 0 & \wedge & \Phi_{i,j,k}\Phi_{i,j,k-1} > 0. & \end{array} \tag{5.60}$$

Assume that after the interface advection the Level-Set function $\Phi$ and the Volume-of-Fluid function $\Psi$ are known. The Volume-of-Fluid function is mass-conserving, but contains unwanted values, i.e. $\Psi < 0$ and $\Psi > 1$ and 'numerical vapor'.

If $\Psi$ contains unwanted values then modified functions $\hat{\Psi}$ and $\hat{\Phi}$ are sought such that:

- $0 < \hat{\Psi} < 0$ only if an interface exists according to $\hat{\Phi}$,
- $\hat{\Psi}$ conserves mass,
- $\hat{\Phi}$ is close to $\Phi$ in order not to destroy local accuracy.

The first constraint would suggest to just remove the unwanted values from $\Psi$, i.e. set $\hat{\Psi}$ to 0 or 1 if no interface is present and limit the function $\hat{\Psi}$ to $0 \leq \hat{\Psi} \leq 1$. However, this would violate the second constraint since mass errors $\epsilon(\mathbf{x})$ arise:

$$\epsilon = \hat{\Psi} - \Psi. \tag{5.61}$$

The total mass lost is then, of course, $\int\limits_{\Omega} \epsilon \, \mathrm{d}\Omega$. The idea is now to first *redistribute* $\epsilon$ to $\hat{\epsilon}$ conservatively and then to add it back to $\Psi$ to obtain the redistributed $\hat{\Psi}$ (which automatically conserves mass):

$$\hat{\Psi} = \Psi + \hat{\epsilon}. \tag{5.62}$$

The new $\hat{\Psi}$ has to satisfy all three constraints.

From the foregoing constraints it follows that a procedure is needed to *redistribute* $\epsilon$ to $\hat{\epsilon}$, such that:

- $\hat{\Psi}$ does not contain unwanted values according to $\hat{\Phi}$,
- the total amount of $\epsilon$ is conserved (in order to conserve mass),
- $\|\hat{\Phi} - \Phi\|$ is small, so that $\hat{\Phi}$ will be close to $\Phi$.

The last remark would suggest to formulate a constrained minimization problem. We choose not to proceed in this direction. Instead, $\epsilon$ will be redistributed in the direction of the nearest interface, to keep the redistributing effect on $\Phi$ local and thus minimizing $\|\hat{\Phi} - \Phi\|$.

Note that $\hat{\Psi}$ and $\hat{\Phi}$ are coupled, which means that changing $\Psi$ will cause changes in $\Phi$, complicating matters even further. A decoupled approach (by means of Picard iterations) will therefore be chosen:

1. keep $\Phi$ fixed, i.e. fix the interface position
2. redistribute $\Psi$ to $\hat{\Psi}$ by:
    (a) define a vector field $\mathbf{q}$ which points towards the interface,
    (b) with $\mathbf{q}$, use a PDE to transport $\epsilon$ towards the interface,
3. with $\hat{\Psi}$ compute $\Phi$ in the usual fashion by the 'inverse function' (see previous section),
4. repeat the procedure (with the updated interface position) when still unwanted values of $\hat{\Psi}$ according to $\hat{\Phi}$ exist.

**2a velocity field q:**

A velocity field $\mathbf{q}(\mathbf{x})$ is needed, which transports ('fluxes') the errors $\epsilon(\mathbf{x})$ towards the interface. In other words, it has to point towards the interface. Such a velocity field can be defined by taking $\mathbf{q} = (u, v, w)^t$:

$$u_{i+\frac{1}{2}} = \begin{cases} 1, & \theta_{i+\frac{1}{2}} < \frac{1}{2} \\ 0, & \theta_{i+\frac{1}{2}} = \frac{1}{2} \\ -1, & \theta_{i+\frac{1}{2}} > \frac{1}{2}, \end{cases} \tag{5.63}$$

where $\theta_{i+\frac{1}{2}}$ is the relative distance of the interface between nodes $i$ and $i+1$, measured from node $i$ (see Eqn. (4.18)):

$$\theta_{i+\frac{1}{2}} = \frac{-\Phi_i}{\Phi_{i+1} - \Phi_i}. \tag{5.64}$$

The magnitude of the components of $\mathbf{q}$ are 1 for reasons of computational efficiency. After some mathematical manipulations, $u$ is found as:

$$u_{i+\frac{1}{2}} = -\mathrm{sign}(|\Phi_{i+1}| - |\Phi_i|). \tag{5.65}$$

and similar in the other coordinate directions.

Problems arise when $|\Phi_i| \approx |\Phi_{i+1}|$, since numerical errors in $\Phi$ will cause a flip-flop effect. Therefore:

$$u_{i+\frac{1}{2}} = 0, \qquad ||\Phi_{i+1}| - |\Phi_i|| < \mathrm{tol}\ \max(|\Phi_{i+\frac{1}{2}}|, h), \tag{5.66}$$

where $\Phi_{i+\frac{1}{2}}$ is approximated by $\frac{1}{2}(\Phi_i + \Phi_{i+1})$, $h$ is the mesh size and tol is a tolerance, typically 0.1.

**2b transport of $\epsilon$:**

With velocity field $\mathbf{q}(\mathbf{x})$, the errors are transported by (for some artificial time $\tau$):

$$\frac{\partial \epsilon}{\partial \tau} + \nabla \cdot (\mathbf{q}\epsilon) = 0, \tag{5.67}$$

until steady state ($\frac{\partial \epsilon}{\partial \tau} = 0$). Eqn. (5.67) is discretized by an upwind approximation in the usual fashion. Instead of taking $\tau \to \infty$, which is rather impractical, after each iteration of Eqn. (5.67) the number of cells (num) with $|\epsilon| > \text{tol}$ is computed. The iterations are stopped if

$$\left\| \frac{\partial \epsilon}{\partial \tau} \right\|_{\infty} \leq \text{tol} \;\vee\; \text{num} = 0 \;\vee\; \text{iter} > \text{maxiter}, \tag{5.68}$$

where tol is a tolerance, typically $10^{-2}$, iter is the iteration-number and maxiter is the maximum number of iterations. The following observations are made:

- the number of cells containing unwanted values of $\Psi$ is small, typically less than 10 for a $64^3$ grid
- the number of iterations of Eqn. (5.67) is small, typically less than 5
- the number of Picard iterations is small, typically 1.

## 5.5 Conclusion

The interface representation and advection is based on the Level-Set methodology. A Volume-of-Fluid function is used to conserve mass. An explicit relationship exists between the Volume-of-Fluid function and the Level-Set function. This relation makes the advection of the Volume-of-Fluid function straightforward. Operator splitting is used for the advection of the Volume-of-Fluid function. Unphysical values of the Volume-of-Fluid function that are due to operator splitting are addressed by mass redistribution, where erroneous mass is transported towards the nearest interface location.

# Chapter 6

# Improved re-initialization

## 6.1 Introduction

Since the interface forces are regularized near the interface, it is necessary that the Level-Set function is a distance function at all time instances. This ensures that the regularization width has finite thickness at all time. This is achieved by the re-initialization procedure of Sussman et al. [37]. The Level-Set function $\Phi$ is re-initialized to make it a distance function by solving until steady state for artificial time $t'$ (see Section 5.3 and Eqn. (5.6))

$$\frac{\partial \Phi}{\partial t'} = N(\Phi, \Phi^0), \tag{6.1}$$

with initial conditions

$$\Phi(\mathbf{x}, 0) = \Phi^0(\mathbf{x}). \tag{6.2}$$

Here $\Phi^0$ defines the interface position after the application of Eqn. (3.11). A first-order temporal discretization of Eqn. (6.1) is:

$$\frac{\Phi^{k+1} - \Phi^k}{\Delta t'} = N_h(\Phi^k, \Phi^0), \tag{6.3}$$

where $N_h$ is defined by Eqn. (5.7).

Due to re-initialization, the interface position can shift considerably (see Van der Pijl et al. [60]: 2D advection tests). This is unwanted, since the MCLS method is based on finding *small corrections* to $\Phi$ in order to conserve mass. Especially small droplets or bubbles (of size about one mesh width) could completely disappear due to re-initialization.

The effects of re-initialization is illustrated by considering a two-dimensional square domain $(x, y) \in [-\frac{1}{2}, \frac{1}{2}]^2$ . There are two interfaces: a straight line at $x = -\frac{1}{4}$ and a circle with the origin at $(x, y) = (0, 0)$ and a diameter of 4 mesh widths. The Level-Set function is initialized such that it is a distance function (see Fig. 6.2(a)). The corresponding Volume-of-Fluid function $\Psi$ is depicted in Fig. 6.2(b). The Level-Set function after re-initialization is shown in Fig. 6.2(c).

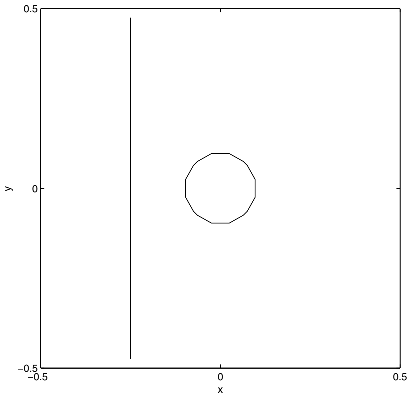


Figure 6.1: Prescribed interfaces on a 20 × 20 mesh

Although the initial Level-Set function is a distance function, large differences exist between the initial and re-initialized Level-Set functions near the circular interface. In fact, the circle has completely disappeared. Consequently, when corrections are made to the Level-Set function in order to satisfy mass (prescribed by the original Volume-of-Fluid function) the corrections are large (see Fig. 6.2(d)). The resulting Level-Set function is highly non-smooth, which is an unwanted effect in our approach.

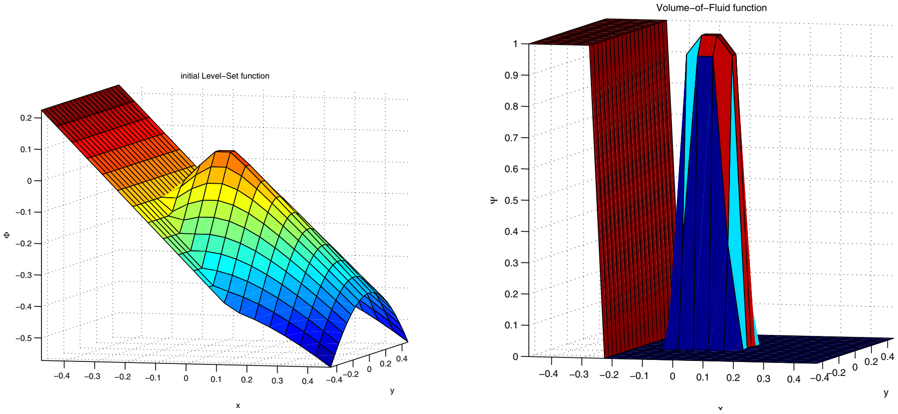
## 6.2 Modified re-initialization

As the previous example has shown, situations exist where the re-initialization procedure of Sussman et al. [37] does not satisfy the needs of the MCLS approach. Various improvements of the original re-initialization procedure exist, often combined with higher order discretization. Since mass is conserved by making corrections after re-initialization, it is not required that the interface maintains the initial position exactly. Besides that, the initial data already resembles a distance function. This leads to a simple modification of the original procedure that satisfies the needs of the MCLS approach.

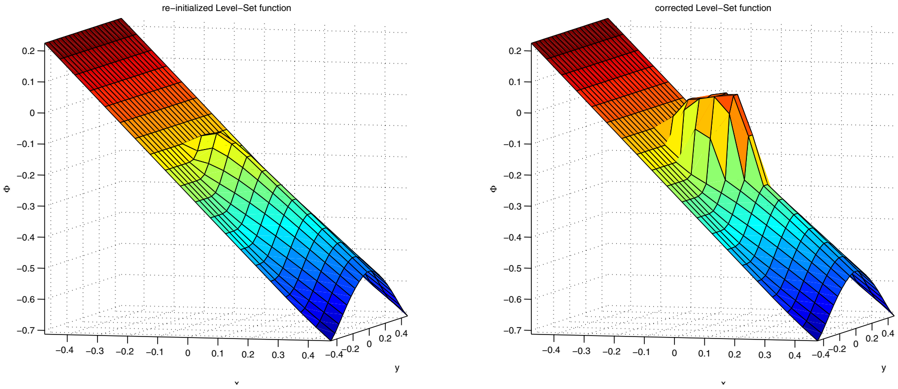
The difficulty with re-initialization is that although it is wanted that that interface position remains intact, no ($\Phi = 0$) boundary conditions are imposed at the interface. The interface is only defined in the initial conditions. The idea is now to leave the values of $\Phi$ near the interface ($\Phi = 0$) unchanged if $-b \leq \Phi \leq b$, where $b \geq 0$ is a prescribed band-width, proportional to the mesh width. This is achieved by modification of Eqn. (6.3):

$$\frac{\Phi^{k+1} - \Phi^k}{\Delta t'} = N_h(\Phi^k, \Phi^0)(1 - q(\Phi^0)) + \frac{\Phi^0 - \Phi^k}{\Delta t'} q(\Phi^0), \tag{6.4}$$

where $q$ is a function, which has value 1 near the interface and value 0 elsewhere.

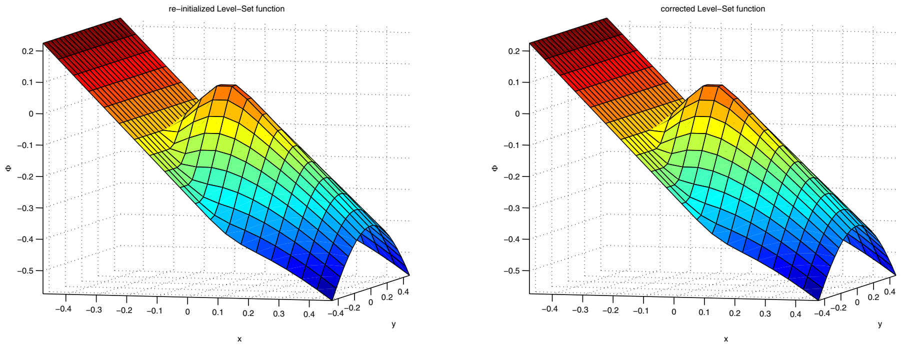


(a) Initial Level-Set function (b) corresponding Volume-of-Fluid function



(c) Level-Set function after re-initialization (d) corrected Level-Set function

Figure 6.2: Effect of re-initialization on small features


re–initialized Level–Set function

corrected Level–Set function

(a) Level-Set function after modified re-initialization

(b) corrected Level-Set function

Figure 6.3: Effect of modified re-initialization on small entities

Note Eqn. (6.4) ensures that $\Phi^{k+1} = \Phi^0$ when $q = 1$ and effectively interface conditions are applied. A logical choice for $q$ would be

$$q(\Phi) = \begin{cases} 1 & |\Phi| \leq b \\ 0 & |\Phi| > b. \end{cases} \tag{6.5}$$

Note that with this approach effectively *boundary conditions* are applied at the $(\Phi^0(\mathbf{x}) = -b)$ and $(\Phi^0(\mathbf{x}) = b)$ contours respectively. The disadvantage of this approach is that $\Phi$ can become non-smooth at $|\Phi| = b$. Therefore, a smooth function $f$ is chosen which mimics the step-wise behavior:

$$q(\Phi^0) = \exp(-(\frac{\Phi^0}{\alpha})^2), \tag{6.6}$$

where $\alpha = \sqrt{\frac{2}{3}(\Delta x^2 + \Delta y^2 + \Delta z^2)}$ is a constant.

The effects of the modification of the re-initialization procedure on the previous described test-case is shown in Fig. 6.3. It can be seen that the re-initialized Level-Set function resembles the initial Level-Set function. The same holds for the corrected Level-Set function, which was the objective of the modification of the re-initialization.

## 6.3 Stop criterion

The iterations of Eqn. (6.4) are stopped if

$$\left\| \frac{\Phi^{k+1} - \Phi^k}{\Delta t'} \right\|_\infty \leq \text{tol}, \tag{6.7}$$

where tol is a tolerance. Note that this quantity acts as the residual of

$$N_h(\Phi, \Phi^0)(1 - f(\Phi^0)) + \frac{\Phi^0 - \Phi}{\Delta t'} f(\Phi^0) = 0, \tag{6.8}$$

so

$$\left\{\begin{array}{rcll} \|N_h(\Phi,\Phi^0)\|_\infty & \leq & \text{tol}, & f(\Phi^0)=0 \\ \|\Phi^0-\Phi\|_\infty & \leq & \text{tol}\ \Delta t', & f(\Phi^0)=1. \end{array}\right. \tag{6.9}$$

We use tol = 0.1 for our tolerance.

## 6.4 Conclusion

Re-initialization is used to make the Level-Set function a distance function. However, the interface can shift considerably. This is due to the fact that the interface is only prescribed in the initial conditions. A simple modification is introduced that suits this research. It uses the fact that the initial condition is already close to a distance function.

# Chapter 7

# Surface tension

## 7.1 Introduction

For the modeling of surface tension effects on the fluid motion, the CSF method of Brackbill et al. [6] is adopted, see Section 2.6 and Section 4.3 for the numerical aspects. With this method the interface forces are transformed into volume forces and added to the right-hand side of Eqn. (2.2). Due to their singular behavior, these surface tension forces are regularized, such that they are distributed over a small region around the interface. According to Eqn. (4.6), these forces are in case of the Level-Set methodology

$$\frac{1}{\rho}\mathbf{f}_s = \frac{1}{\frac{1}{2}\left(\rho_0 + \rho_1\right)}\ \sigma\kappa\delta_\alpha(\Phi)\nabla\Phi. \tag{7.1}$$

where $\delta_\alpha$ is a regularized Dirac delta function, $d$ is the nearest distance to the interface, $\rho_0$ is the density of fluid '0', $\rho_1$ the density of fluid '1', $\sigma$ is the surface tension coefficient and $\kappa$ is the curvature of the interface.

## 7.2 Discretization of curvature

The curvature $\kappa$ is defined by

$$\kappa = \nabla \cdot \frac{\nabla\Phi}{|\nabla\Phi|}. \tag{7.2}$$

For the ease of implementation, this is rewritten as (see e.g. Brackbill et al. [6], Rudman [26], Kang et al. [58]):

$$\kappa = \frac{\nabla\cdot\nabla\Phi}{|\nabla\Phi|} - \frac{\nabla\Phi\cdot[(\nabla\Phi\cdot\nabla)\nabla\Phi]}{|\nabla\Phi|^3}, \tag{7.3}$$

which is a symbolic notation for:

$$\kappa = \frac{\nabla\cdot\nabla\Phi}{|\nabla\Phi|} - \frac{1}{|\nabla\Phi|^3}\sum_i\sum_j \frac{\partial\Phi}{\partial x_i}\frac{\partial\Phi}{\partial x_j}\frac{\partial^2\Phi}{\partial x_i\partial x_j}. \tag{7.4}$$

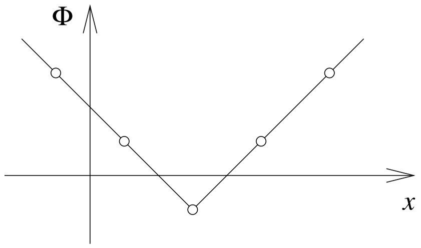


Figure 7.1: Local extremum of the Level-Set function $\Phi$ between two approaching interfaces

The second order derivatives are discretized by straightforward central differencing. The first order derivatives however need special care. It is immediately visible that problems arise in Eqn. (7.4) when $|\nabla\Phi|$ vanishes. Such a situation occurs when the Level-Set function has a local extremum. This happens when two interfaces are approaching each other, see Fig. 7.1 for an example. To overcome this, instead of using central differences, the first order derivatives are discretized by taking the maximum (in modulus) of the left-hand-side, right-hand-side and central scheme. This is a first order approximation. It is important when droplets or bubbles become small and when droplets or bubbles merge. It also increases the smoothness of the curvature distribution.

## 7.3 Parasitic currents

The difficulty with the CSF method is that so-called parasitic currents can occur. These are small unphysical currents that according to Lafaurie et al. [20] are due to 'slight unbalance of stresses at the sites in the interfacial region'. Parasitic currents have been studied for a Volume-of-Fluid method by Scardovelli and Zaleski in [27] and by Lafaurie et al. in [20]. Tryggvason et al. have analyzed parasitic currents in [4] for a front-tracking method and Enright et al. in [16] for a Level-Set method.

The parasitic currents are caused by the distribution of interface forces and by inaccuracies in the computation of the curvature. In case of the Level-Set methodology, the curvature $\kappa$ is given by Eqn. (7.4). which is approximated by central differences. Computing the curvature with pure Level-Set methods faces no problems because of the smoothness of the Level-Set function. However, with the MCLS method additional difficulties are encountered. With the MCLS approach corrections to the Level-Set function are made *locally* near the interface in order to conserve mass. This results in irregular data near the interface. From Eqn. (7.4) follows that it will cause local errors in $\kappa$ of the order $\frac{\epsilon}{h^2}$, where $\epsilon$ is the magnitude of the corrections and $h$ is the mesh width. Although the corrections are small in general, situations might occur where they are $\mathcal{O}(h)$ locally. See Fig. 6.2(d) for an example with unmodified re-initialization. Note that in this case the errors are completely due to re-initialization, since no advection is performed in this example. The consequences for the curvature are rather dramatic in the sense that parasitic currents will grow when the mesh is refined. The order of the errors in $\kappa$ is $\frac{1}{h}$. A study of these phenomena is performed by Coyajee et al. in [61].

The Laplace test-case is frequently used to study the parasitic currents, see e.g. Rudman [26], Lafaurie et al. [20], Tryggvason et al. [4]. A sphere with radius $\frac{1}{4}m$ is placed in the center of a cubic domain with dimensions $1 \times 1 \times 1m$. The flow is initially at rest. Since the initial conditions satisfy the steady-state problem, all velocities are parasitic currents. The following gravity and material constants are used: $g = 0\ \frac{m}{s^2}$, $\sigma = 0.01\ \frac{kg}{s^2}$, $\rho_0 = 1\ \frac{kg}{m^3}$, $\frac{\rho_1}{\rho_0} = 1$, $\mu_0 = 0.1\ \frac{kg}{ms}$ and $\frac{\mu_1}{\mu_0} = 1$. In Fig. 7.2 the time evolution of parasitic currents are shown for three different grid sizes: $32^3$, $64^3$ and $96^3$. The maximum values (in modulus) of the parasitic currents grow in time and become larger for increasing mesh size. The grid dependency is $\mathcal{O}\left(\frac{1}{h}\right)$. The currents are shown in Fig. 7.3

Our aim is now to reduce the parasitic currents to a small value that is negligible compared to the physical currents and does not increase when the mesh is refined. This is achieved by:

- reduction of the magnitude of the corrections caused by re-initialization
- smoothing of the curvature

## 7.4 Re-initialization start criterion

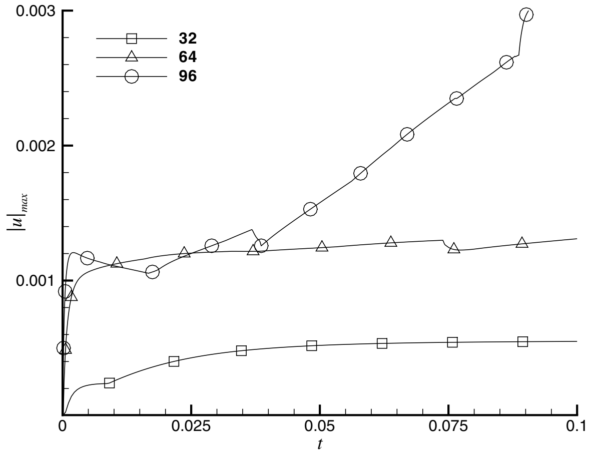
The re-initialization procedure introduces errors in the Level-Set function $\Phi$. It is unwanted that these errors are larger than the changes per time-step due to advection of $\Phi$ (by means of Eqn. (3.11)). The re-initialization procedure should therefore only be carried out when necessary. The Level-Set function is not re-initialized each time-step but only when the interface has traveled some distance. This is measured by a time-step $\Delta t_{\text{reinit}}$:

$$\Delta t_{\text{reinit}} = \frac{\sigma}{\frac{||u||_\infty}{\Delta x} + \frac{||v||_\infty}{\Delta y} + \frac{||w||_\infty}{\Delta z}}, \tag{7.5}$$

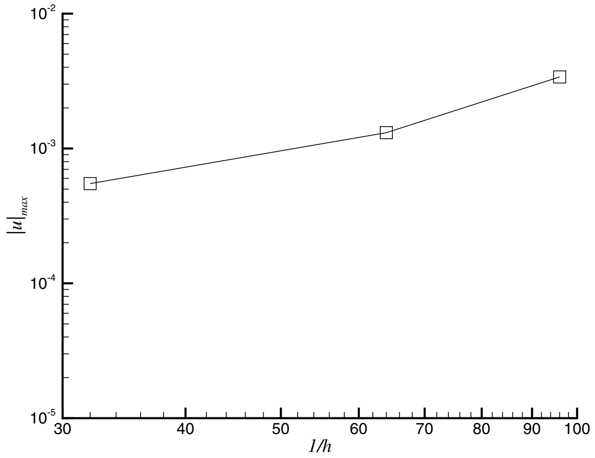
where $\sigma = 0.1$ is chosen. Re-initialization is only performed if $t^n - \Delta t_{\text{reinit}}$ is larger than or equal to the last re-initialization time. The effect of the time-step criterion on the parasitic currents is shown in Fig. 7.4. The maximum modulus of the parasitic currents converge to a constant value for all grids. The effect on the $96^3$ grid is most significant.

## 7.5 Effect of mass redistribution

The previous figures showed that the velocities (parasitic currents) are very small for the Laplace test case. This means that the displacement of the interface



(a) time evolution; The maximum modulus of the parasitic currents does not converge to a constant value for all grids



(b) grid dependency at $t = 0.1$

Figure 7.2: Parasitic currents for the Laplace test-case with three different grids

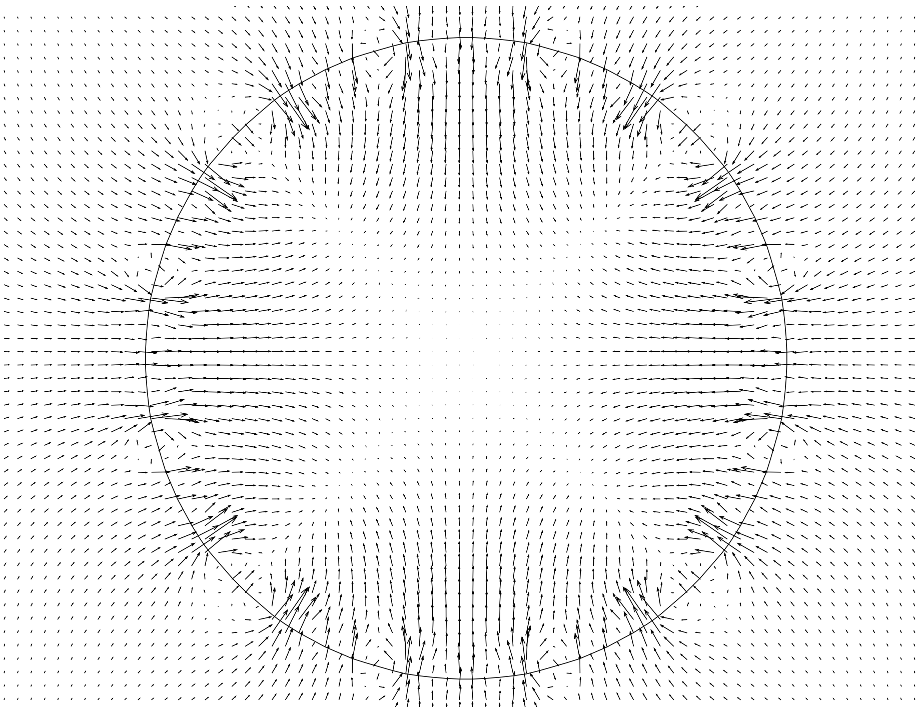
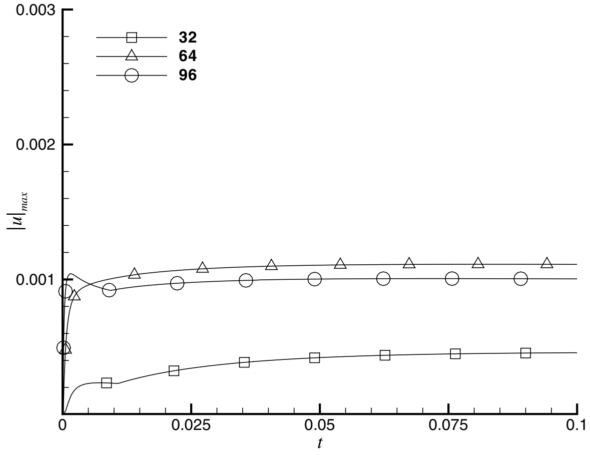
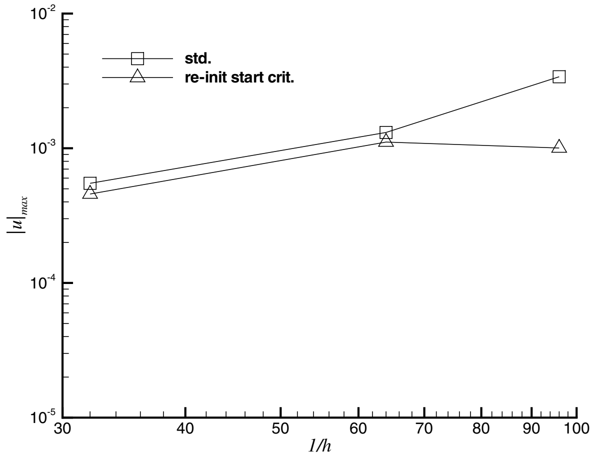


Figure 7.3: Parasitic currents for the Laplace test-case in the symmetry plane; $96^3$ grid



(a) time evolution; The maximum modulus of the parasitic currents converge to a constant value for all grids



(b) grid dependency at $t = 0.1$; Note that the case without the start criterion ('std') is also include in this figure; The parasitic currents decrease due to the start criterion, most notably for the $96^3$ grid

Figure 7.4: Parasitic currents for the Laplace test-case with three different grids; start criterion in re-initialization

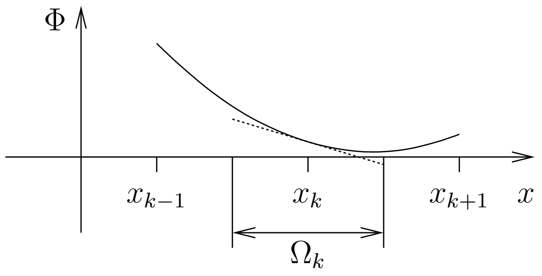


Figure 7.5: One-dimensional example of a Level-Set function $\Phi$ that does not change sign in $\Omega_k$ (hence no interface); but the linearization of $\Phi$ around $\mathbf{x}_k$ does, hence $0 < \Psi_k < 1$ and the classification 'numerical vapor'.

is equally small. It can therefore be expected that the Level-Set function $\Phi^*$ after pure Level-Set advection (Eqn. (5.5)) and without re-initialization does not loose a significant amount of mass. At first sight, there would by no need for any corrections on $\Phi^*$ in order to conserve mass. However Fig. 7.4 shows that the parasitic currents do not decrease when the grid is refined, which indicates that still corrections are made that influence the computation of curvature by means of Eqn. (7.4). Further inspection of the numerical output reveals that the corrections are due to mass redistribution (see Section 5.4.4). This means that 'numerical vapor' exists in the initial conditions according to the mass-distribution criterion as specified by Eqn. (5.60). In other words, a situation exists where the Level-Set function $\Phi$ in a grid point $\mathbf{x}_{i,j,k}$ has the same sign as $\Phi$ in its neighbors $\mathbf{x}_{i\pm1,j,k}$, $\mathbf{x}_{i,j\pm1,k}$ and $\mathbf{x}_{i,j,k\pm1}$, but still an interface exists in the computational cell $\Omega_k$ according to the linearization of $\Phi$ around $\mathbf{x}_k$. A one-dimensional example is shown in Fig. 7.5. The mass redistribution will redistribute all mass in $\Omega_{i,j,k}$ in the direction of the nearest interface and a new Volume-of-Fluid function $\Psi$ is obtained (see Section 5.4.4). With the new Volume-of-Fluid function $\Psi$, corrections are made to the Level-Set function $\Phi$ such that it satisfies $f(\Phi, \nabla\Phi) = \Psi$, as explained in Section 5.4.3.

Fig. 7.6 shows the parasitic currents without mass redistribution. No corrections to $\Phi^*$ are made and the pure Level-Set approach is obtained. The parasitic currents are completely due to the regularization of the interface forces (CSF approach, see Sections 2.6 end 4.3) and converge with $\mathcal{O}(h)$. This is the pure Level-Set behavior and shows that the parasitic currents are due to mass redistribution.

The Laplace test case is believed to be the worst-case scenario for the 'Mass Conserving Level-Set' approach. On the one hand corrections are made due to mass redistribution and the initial Level-Set function becomes perturbed. On the other hand the interface does not move and the perturbations remain intact since $\frac{\partial\Phi}{\partial t} = -\mathbf{u}\cdot\nabla\Phi \approx 0$. This means that the perturbations will never disappear. When the interface *does* move in practical applications, the perturbations will (numerically) diffuse after the convection step (Eqn. (5.5)). It is then advantageous to compute the curvature before the corrections are made. Once again, this makes no difference for the Laplace test case since the interface is stationary.

## 7.6 Curvature smoothing

Volume-of-Fluid methods encounter similar, but more severe difficulties when computing the curvature, since the Volume-of-Fluid function has a step-like behavior. With these methods it is common fashion to effectively compute the curvature based on a smoothed function, see e.g. Scardovelli and Zaleski [27] or Rudman [26]. With front-tracking methods Sousa et al. [2] remove 'undulations' from the interface before computing the curvature. Doing so, they effectively smooth the interface grid. Both approaches have been an inspiration for this work.

The first step that comes to mind would be to smooth the Level-Set function before computing the curvature. However, a smooth distribution of the *curvature* is sought, not of the Level-Set function itself. Therefore, it is chosen to smooth the curvature instead. This should not be performed carelessly, since some properties of the curvature should be maintained. These are:

1. the curvature is continuous and sufficiently smooth,
2. the curvature $\kappa$ is the divergence of $\dfrac{\nabla\Phi}{|\nabla\Phi|}$, so $\kappa$ has to be modified conservatively,
3. the curvature has to correspond to the interface at the interface ($\Phi = 0$) when the curvature is already sufficiently smooth.

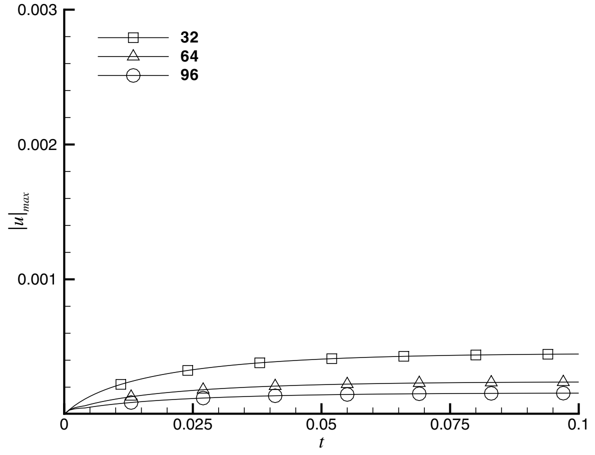
To meet these demands, an approach based on a diffusion equation in pseudo-time $\tau$ is chosen:

$$\frac{\partial\kappa}{\partial\tau} = \nabla \cdot d\nabla\kappa. \tag{7.6}$$

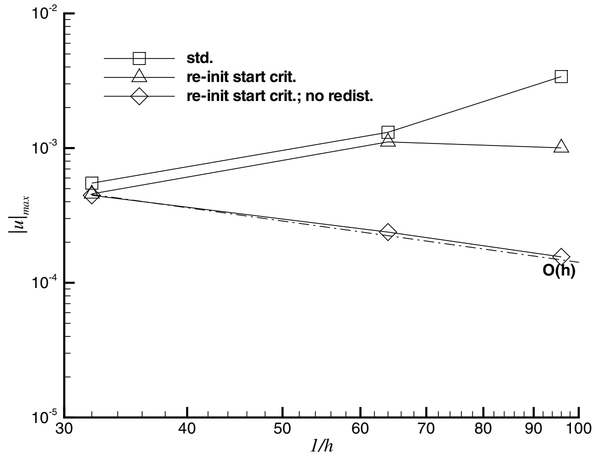
The first two requirements are satisfied if Eqn. (7.6) is used provided $d$ is smooth. Here $d$ is a diffusion coefficient that ensures that the last demand is also met. It is constructed in the following way:

$$d = 1 - \exp\left(\frac{-\Phi^2}{\Delta x^2 + \Delta y^2 + \Delta z^2}\right). \tag{7.7}$$

Note that $d$ is smooth. Since the diffusion coefficient is very small near the interface, significant modifications to $\kappa$ are only made when the derivatives $\nabla\kappa$ are large and $\kappa$ is left intact when it is smooth. An explicit Euler temporal discretization is used. The maximum allowed pseudo-time step (for $d \leq 1$) of $\frac{h^2}{6}$ is used. For the time-step of the explicit Euler method $\frac{N}{16}$ is used, where $N$ is the number of grid cells in one direction. Note that the number of iterations is proportional to the mesh size.

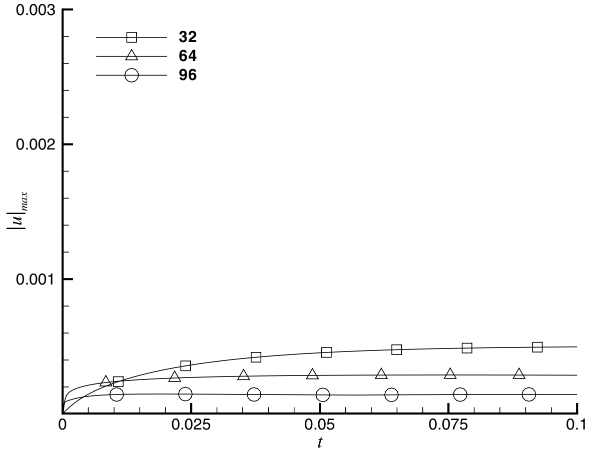


(a) time evolution; The maximum modulus of the parasitic currents converge to a constant value for all grids

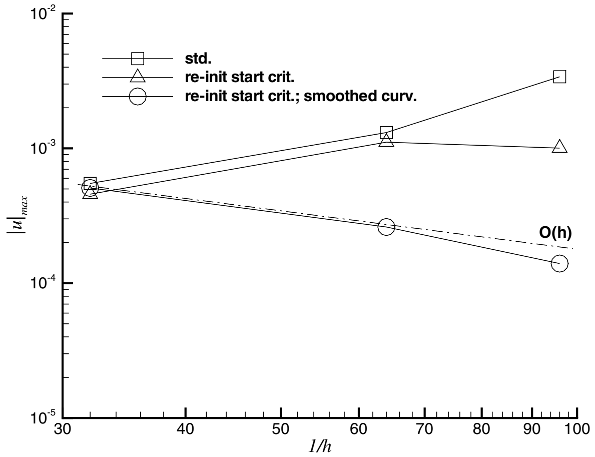


(b) grid dependency at $t = 0.1$; Note that the two previous cases ('std' and 're-init start crit.') are also included in this figure

Figure 7.6: Parasitic currents for the Laplace test-case with three different grids; start criterion in re-initialization and no mass redistribution



(a) time evolution; The maximum modulus of the parasitic currents converge to a constant value for all grids



(b) grid dependency at $t = 0.1$; Note that the two previous cases ('std' and 're-init start crit.') are also included in this figure; The parasitic currents decrease further due to smoothing of the curvature

Figure 7.7: Parasitic currents for the Laplace test-case with three different grids; time-step criterion in re-initialization and smoothed curvature

The results for the Laplace test-case are shown in Fig. 7.7. The maximum value (in modulus) of the parasitic currents converges with $\mathcal{O}(h)$ as shown in Fig. 7.7(a).

## 7.7 Conclusion

First and second order derivatives of the Level-Set function have to be computed for the modeling of surface tension. Since local corrections to the Level-Set function are made near the interface in order to conserve mass, problems arise when these derivatives are computed. This results in parasitic currents. To reduce these currents, the corrections are made smaller by introducing a start criterion for the re-initialization procedure. Besides that, the curvature is smoothed based on a diffusion equation, which reduces the currents even further.

# Chapter 8

# Applications

## 8.1 Introduction

The behavior of the MCLS approach is shown by a couple of standard advection tests, with a prescribed velocity field. Thereafter, the method is applied to the complete set of equations by considering a falling drop and a rising bubble in two and three dimensions, respectively. Merging of rising bubbles is studied for two aligned and two misaligned bubbles.

## 8.2 Two-dimensional results

### 8.2.1 Linear advection

The first advection test is presented in Fig. 8.1. The velocity field is prescribed by $(u, v) = (0, -1)$. The dimensions of the computational domain are: $L_x = 10$ and $L_y = 100$. We use a 10x100-mesh. Initially a circle of radius $R_0$ is placed at $x = L_x/2$ and $y = L_y - 2R_0$. For the case of $R_0 = 4$ (a circle with a diameter of 8 mesh sizes), the relative mass is plotted in Fig. 8.2 as function of the traversed distance of the circle. First-order, second-order and third-order pure Level-Set simulations (with and without re-initialization) are compared with the MCLS method. ENO discretization is adopted for the pure Level-Set method (see aforementioned references). The order of re-initialization is in agreement with the order of advection. The tolerance in the VOF advection is taken to be: $\epsilon = 10^{-8}$. Globally speaking it can be said that mass is always lost for the pure Level-Set advection. Mass losses are smaller for higher accuracy and re-initialization causes much higher mass losses. The MCLS method conserves mass up to the specified tolerance.

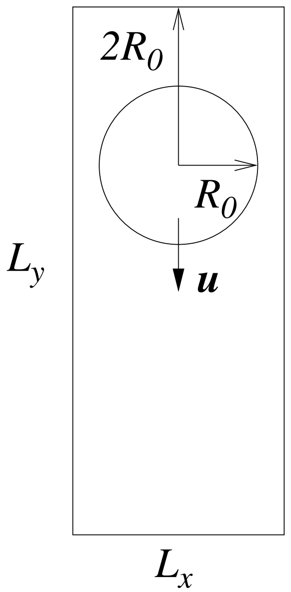


Figure 8.1: Linear advection test. A disc of radius $R_0$ is advected in a rectangular domain of $L_x$ by $L_y$. The center of the disc is initially placed at $2R_0$ from the top of the domain. The advection velocity is $u$.

### 8.2.2 Zalesak's rotating disc

The advection test of Zalesak ([17]) is often used to demonstrate the interface-advection algorithm (see e.g. [24, 25, 29] for VOF methods and [13, 15, 44, 45] for Level-Set methods). A slotted disc (Fig. 8.3) is rotated through one revolution around the center of the computational domain. The velocity-field is prescribed by: $(u, v) = (-(y - \frac{1}{2}L_y), x - \frac{1}{2}L_x)$. The center of the slotted disc $(x_0, y_0)^t$ is located at $(x_0, y_0) = (\frac{1}{2}L_x, \frac{3}{4}L_y)$. The sizes are: $L_x = L_y$, $R_0 = \frac{3}{20}L_x$ and $W = \frac{1}{3}R_0$.

In Figs. 8.4 and 8.5 results are shown for various mesh sizes. As might be expected, given the foregoing linear advection results, mass is still lost with the high-order Level-Set method. For the MCLS method, mass is conserved up to the specified tolerance $\epsilon$, although mass is redistributed due to numerical diffusion. Results of the MCLS method are comparable with VOF/PLIC methods (see aforementioned references). Note that the Level-Set advection is first-order in the case of the MCLS method.

The length $l(S)$ of the interface $S$ can be expressed as (see e.g. [36])

$$l(S) = \int_S \mathrm{d}S = \int_\Omega \delta(\Phi)|\nabla\Phi|\,\mathrm{d}\Omega. \tag{8.1}$$

This is approximated by using central differences and regularization of the Dirac

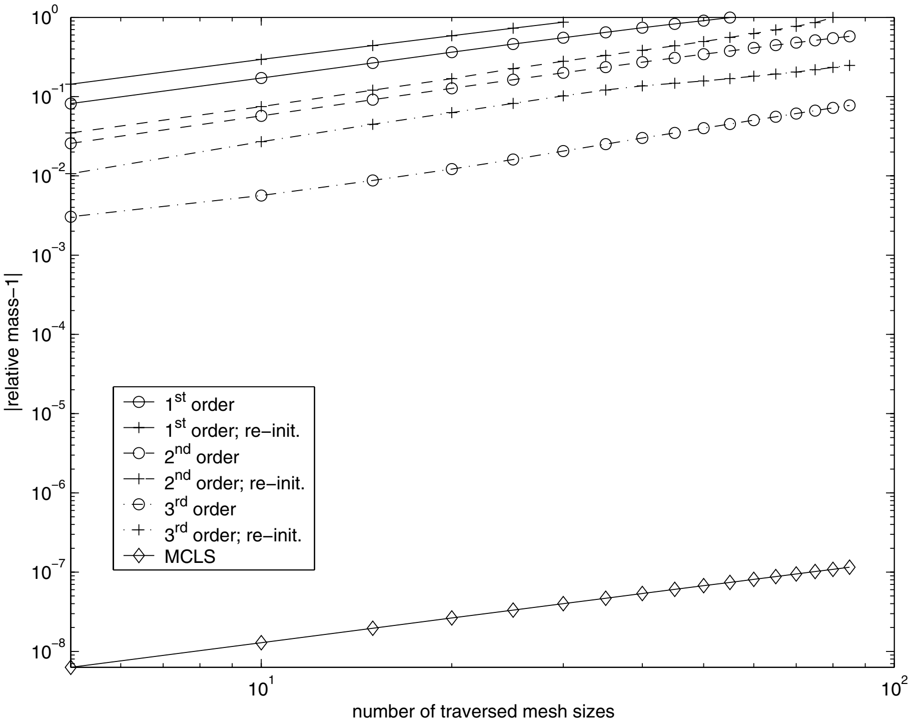


Figure 8.2: Relative mass errors for the linear advection test; $\epsilon = 10^{-8}$ (every $10^{th}$ iteration marked). Pure Level-Set advection with different discretization orders are compared with MCLS.

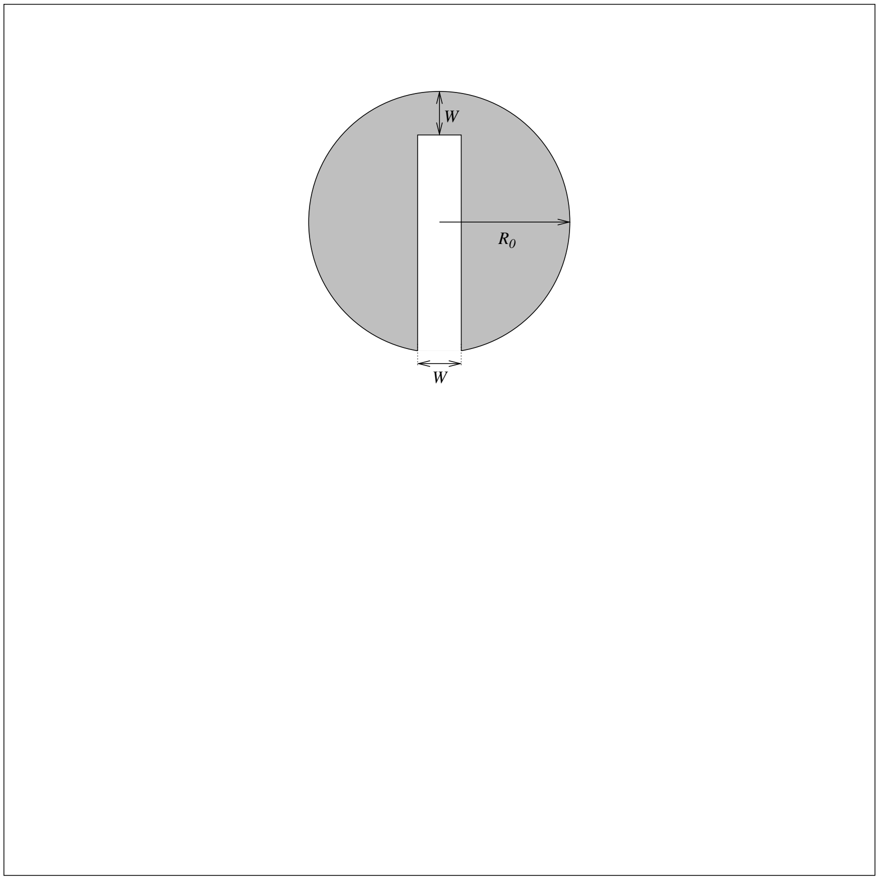


Figure 8.3: Zalesak's slotted disc advection test (to scale). The dimensions of the slot are depicted in the graph.

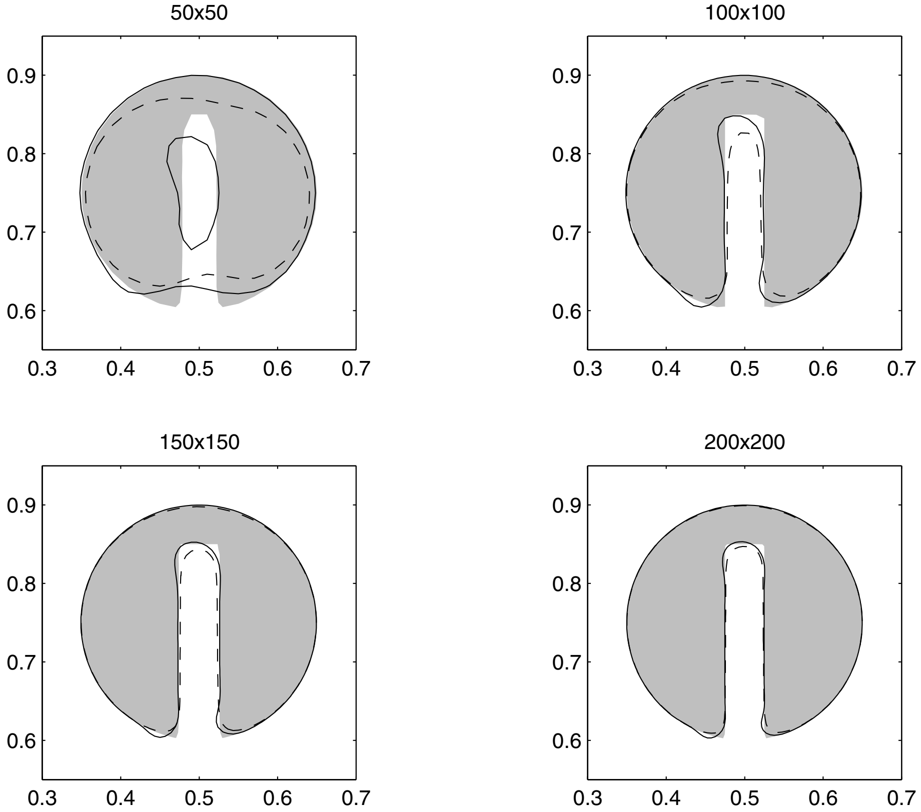


Figure 8.4: Results for Zalesak's advection test; The shaded area indicates the initial contour. The dashed lines indicate the interface after one revolution with $3^{rd}$ order pure Level-Set advection. The solid lines correspond to MCLS advection. Four different mesh sizes have been employed.

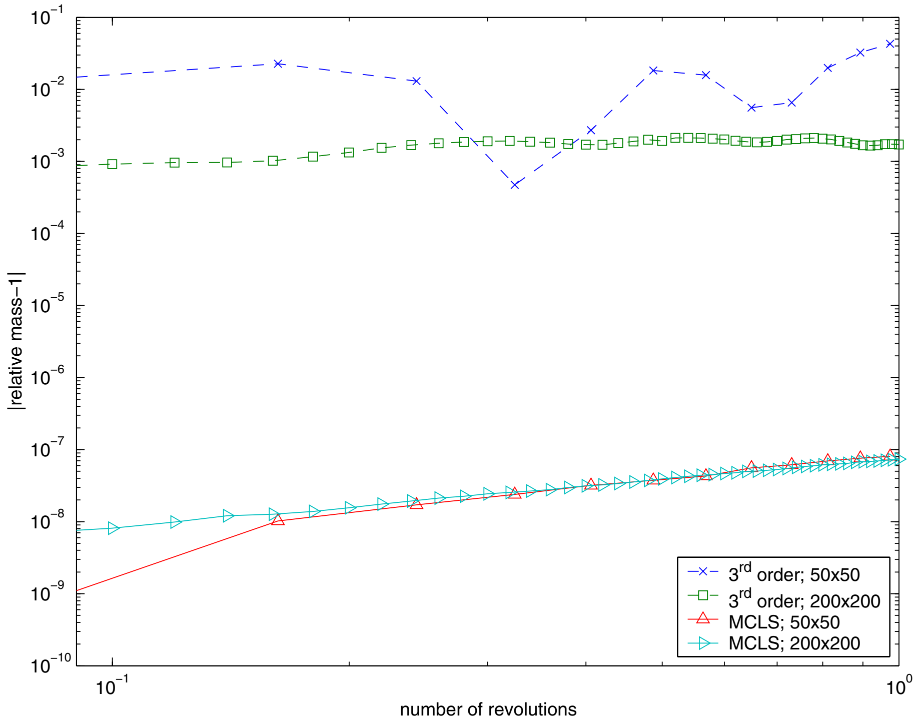


Figure 8.5: Relative masses for Zalesak's advection test; $\epsilon = 10^{-8}$ (every $50^{th}$ iteration marked). Pure Level-Set advection with third order discretization is compared with MCLS for different mesh sizes.

| $\frac{l(S)}{l(S^0)}$ | $50\times 50$ | $100\times 100$ | $150\times 150$ | $200\times 200$ |
|---|---|---|---|---|
| initial | 0.86094 | 0.98187 | 0.98804 | 0.99102 |
| $3^{rd}$ order | 0.49236 | 0.80940 | 0.91253 | 0.93318 |
| MCLS | 0.84106 | 0.95977 | 0.97020 | 0.97570 |

Table 8.1: Computed interface lengths after one revolution

delta function (see Eqn. (4.5)):

$$l(S) = \int_\Omega \delta_\alpha(\Phi)|\nabla\Phi|\,\mathrm{d}\Omega, \tag{8.2}$$

where

$$\delta_\alpha(x) = \begin{cases} 0, & |x| > \alpha \\ \frac{1+\cos(\frac{\pi x}{\alpha})}{2\alpha}, & |x| \le \alpha. \end{cases} \tag{8.3}$$

Note that due to Eqn. (4.5), the exact value of $\Phi$ has no meaning in the Level-Set formulation; only its sign is relevant. The $\alpha$ in Eqn. (8.2) therefore equals the $\alpha$ of Eqn. (4.5). The exact length of the interface is

$$l(S^0) = \left(4 + 2\pi - 2\arctan(\frac{1}{2}\frac{W}{R_0}) - \frac{W}{R_0}\right) R_0. \tag{8.4}$$

In Table 8.1 the computed interface lengths after one revolution are compared with the exact length. ‘Initial’ means at $t = 0$, when errors are made due to the regularization of the delta function. Furthermore, ‘$3^{rd}$ order’ and ‘MCLS’ correspond to the interface lengths after one revolution.

Since $\Phi^0$ is a distance function, $|\Phi - \Phi^0|$ is a measure for the shift of the interface after one revolution. A norm of the error $e$ can be defined as

$$||e||_2 = \left(\frac{\int_S |\frac{\Phi-\Phi^0}{L_x}|^2\,\mathrm{d}S}{\int_S \mathrm{d}S}\right)^{\frac{1}{2}} = \left(\frac{\int_\Omega |\frac{\Phi-\Phi^0}{L_x}|^2\,\delta_\alpha(\Phi)|\nabla\Phi|\,\mathrm{d}\Omega}{\int_\Omega \delta_\alpha(\Phi)|\nabla\Phi|\,\mathrm{d}\Omega}\right)^{\frac{1}{2}}, \tag{8.5}$$

where $L_x$ is used to non-dimensionalize $\Phi$ and $\Phi^0$ is the initial Level-Set function. Results are presented in Fig. 8.6. In all cases the MCLS approach is superior to the $3^{rd}$ order ENO, pure Level-Set method.

### 8.2.3 Air/water flow

In [58] a two-dimensional rising air bubble in water is considered. The dimensions and sizes are: $L_x = 0.02\ m$, $L_y = \frac{3}{2}L_x$, $R_0 = \frac{1}{6}L_x$, $x_0 = y_0 = \frac{1}{2}L_x$. The gravity and material constants are: $g = 9.8\ \frac{m}{s^2}$, $\sigma = 0.0728\ \frac{kg}{s^2}$, $\rho_w = 10^3\ \frac{kg}{m^3}$, $\rho_a = 1.226\ \frac{kg}{m^3}$, $\mu_w = 1.137\ 10^{-3}\ \frac{kg}{ms}$ and $\mu_a = 1.78\ 10^{-5}\ \frac{kg}{ms}$, where subscripts $_w$ and $_a$ indicate water and air respectively.

Results are shown in Fig. 8.7 for three different mesh sizes. We take $\epsilon = 10^{-8}$. Relative mass losses are of the same order and in agreement with the advection

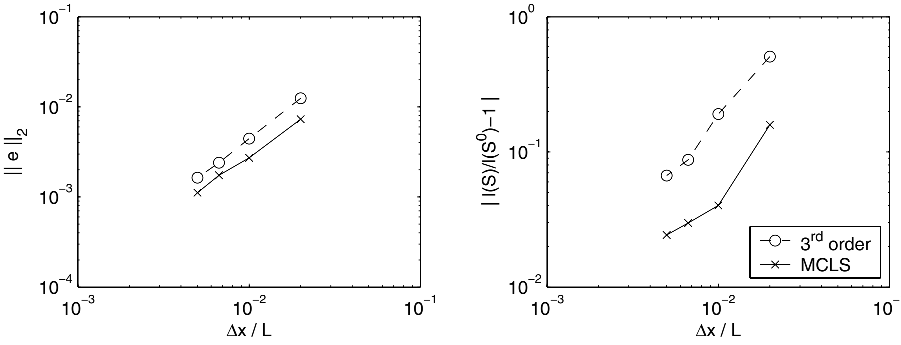


Figure 8.6: Errors for Zalesak's advection test. The left graph show the errors of the Level-Set-function $\Phi$ near the interface after one revolution. The right graph shows the errors in the computed interface length.

tests. Note that the number of grid cells is much smaller than in [58]. The results are the same for $t \leq 0.025$ for all mesh sizes. Thereafter small differences occur. The results compare well with [58]. The MCLS method seems to result in a more compact structure at the highly curved part of the interface at $t = 0.05$. This is thought to be caused by the low resolution of the grids used here.

In Fig. 8.8 results are shown for a falling droplet. The conditions are the same as for the rising bubble, except for the sign of $\Phi$ at $t = 0$ and $y_0 = L_x$. Note that we use homogeneous Neumann boundary conditions for the Level-Set function $\Phi$ and Volume-of-Fluid function $\Psi$. Mass conservation properties are the same as before. The results are the same until the droplet hits the bottom. Thereafter differences occur. This is thought to be due to limited number of grid cells available to capture the flow-phenomena near the wall. The results compare well with [58]. Note that the results in [58] span $t \leq 0.05$; no results after collision are presented.

### 8.2.4 Computational Costs

The computational costs of the MCLS method are compared with purely third order Level-Set advection in Table 8.2. The numbers correspond to CPU seconds spent per time-step for the rising bubble test case on a $60 \times 90$ mesh. Note that the Level-Set advection is first order accurate in case of MCLS. The interface advection for MCLS becomes approximately twice as expensive. This makes the total time-step approximately 10% more expensive. On the other hand, mass is conserved up to the specified tolerance in case of MCLS, whereas mass is lost/gained for the third order pure Level-Set advection (see Figs. 8.2 and 8.5 for the advection tests).

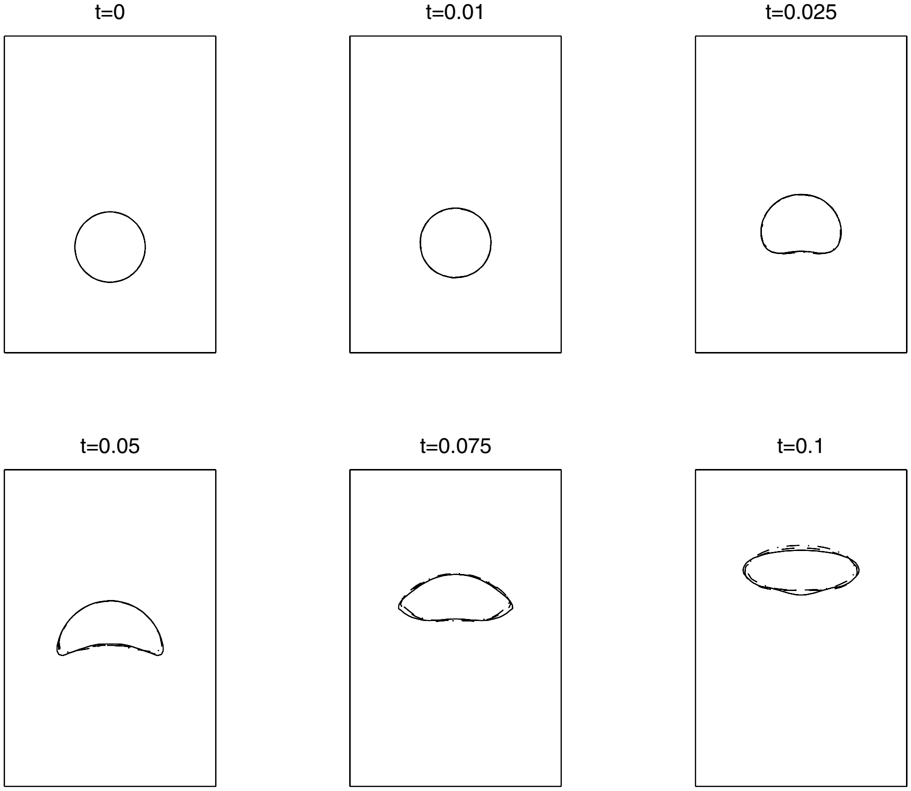


Figure 8.7: Interface positions for the rising bubble. Three different mesh sizes have been employed: $-\cdot-$ : $30\times45$; $--$ : $40\times60$; — : $60\times90$ mesh. Snapshots are presented with equidistant time-steps.

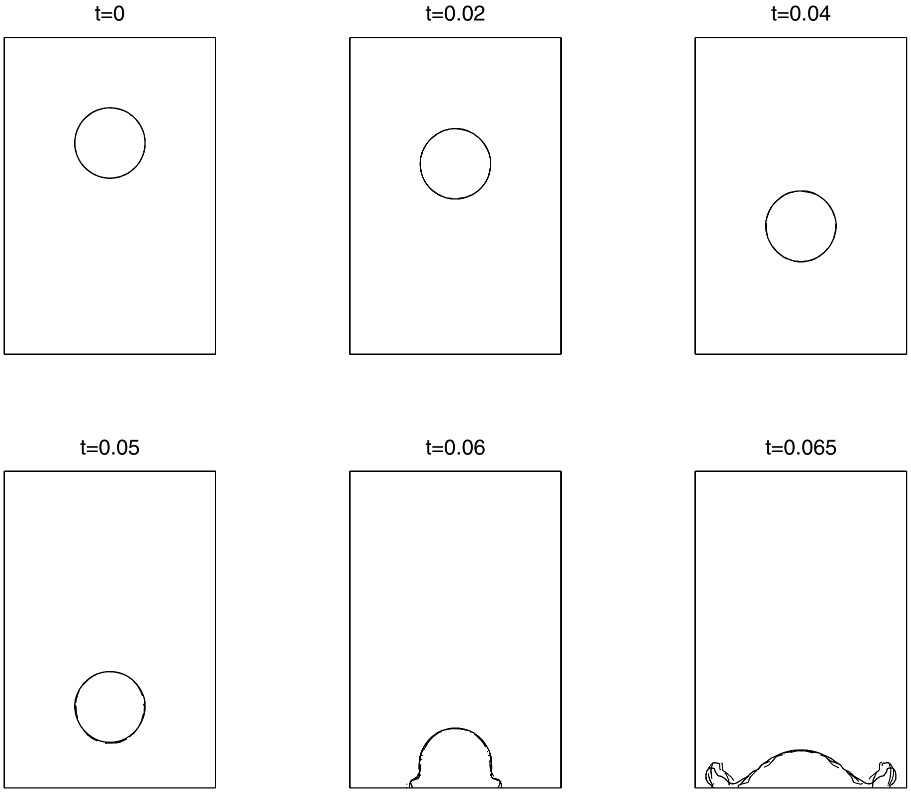


Figure 8.8: Interface positions for the falling droplet; Three different mesh sizes have been employed: $-\cdot-$ : $30 \times 45$; $--$ : $40 \times 60$; — : $60 \times 90$ mesh. Snapshots are presented with equidistant time-steps.

| | MCLS | | $3^{rd}$ order |
|---:|:---|---:|---:|
| Level-Set advection | | 0.03 | 0.07 |
| re-initialization | | 0.02 | 0.07 |
| VOF advection | | | |
| flux $x$ | 0.04 | | |
| correct | 0.04 | | |
| flux $y$ | 0.04 | | |
| correct | 0.05 | | |
| redist | 0.06 | | |
| total | | 0.23 | |
| Total advection | | 0.28 | 0.14 |
| Total time-step | | 1.50 | 1.35 |

Table 8.2: Computational costs measured in CPU seconds per time-step. MCLS is compared with third order Level-Set advection for the rising bubble test case on a $60 \times 90$ mesh.

## 8.3 Three-dimensional results

### 8.3.1 Advection test

Consider a sphere which is placed in the center of a cube. The width of the cube is twice the diameter of the sphere. There are 30 grid cells in $x$, $y$ and $z$-direction. Periodic boundary conditions are applied at the boundaries. The CFL number is 0.35. Note that this CFL number is also used in the full model simulations. The sphere is advected with uniform velocity field $(u, v, w) = (2, 3, -1)$. Relative mass errors are plotted in Fig. 8.9. The results are consistent with the two-dimensional work presented in the previous section.

### 8.3.2 Rising air bubble in water without surface tension

The behavior of the MCLS approach is shown by a rising bubble in three dimensions. The gravity and material constants are:

$$\begin{aligned} g &= 9.8\,\frac{m}{s^2}, & \sigma &= 0.0728\,\frac{kg}{s^2}, \\ \rho_w &= 10^3\,\frac{kg}{m^3}, & \rho_a &= 1.226\,\frac{kg}{m^3}, \\ \mu_w &= 1.137 \times 10^{-3}\,\frac{kg}{ms}, & \mu_a &= 1.78 \times 10^{-5}\,\frac{kg}{ms}, \end{aligned}$$

where subscripts $_w$ and $_a$ indicate water and air respectively. The initial conditions are presented in Fig. 8.10. An air bubble is initially at rest and aligned on the center line of the computational domain. The dimensions of the domain are: $L_x = L_y = L_z = 0.01\ m$. The radius of the bubble is $R = 0.00125\ m$. The distance from the center of the bubble to the floor of the domain is $z_0 = 0.0025\ m$. A free surface is located at $z_1 = 0.0075\ m$. Results are shown in Figs. 8.11 and

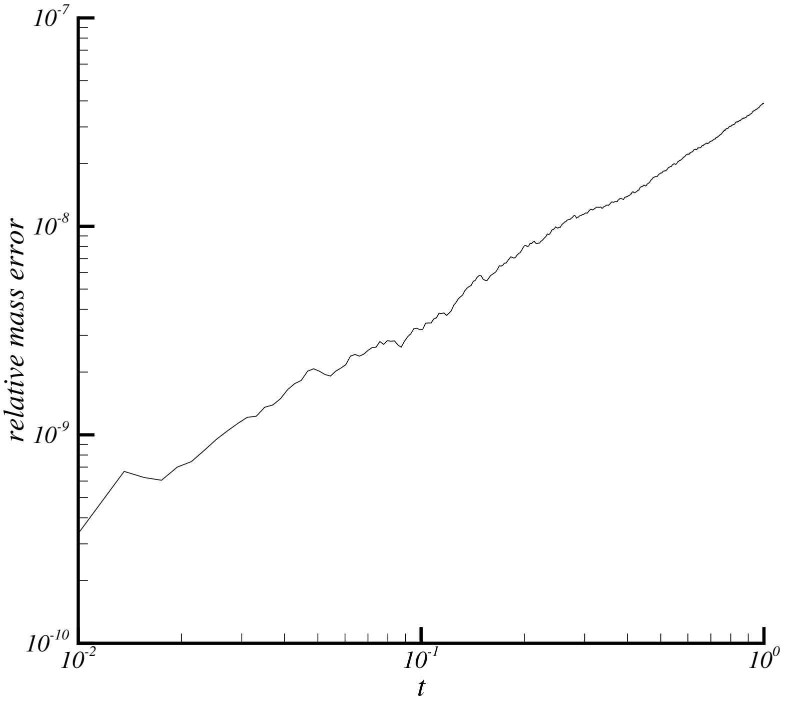


Figure 8.9: Relative mass errors for advected sphere; $\epsilon = 10^{-8}$

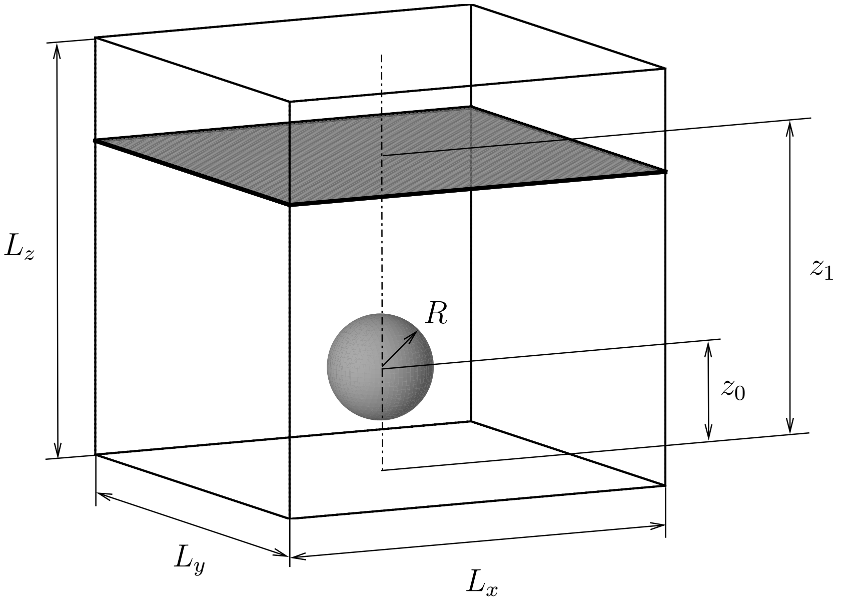


Figure 8.10: Set-up of the rising-bubble test case

8.12. Two grid sizes are considered. These are 64 × 64 × 64 and 96 × 96 × 96 respectively. The snapshots are taken at equal time differences of 0.005 *sec*. For ease of visualization, only $y < \frac{1}{2}\ L_y$ is plotted. Also, the interface position in the plane $y = \frac{1}{2}\ L_y$ is plotted. It can be seen that the bubble deforms and breaks up to form a bell- and ring-like structure, just before it breaks through the free surface. In [1] similar simulations are made for a rising bubble with low surface tension. Good resemblance is observed with these results (see Fig. 8.13) and the snapshots presented in Fig. 8.14.

In Fig. 8.15 the rise speed $w_c$ of the droplet before collision with the free surface is plotted as function of time $t$. For the rise speed $w_c$ the velocity of the center of gravity of the bubble is taken:

$$w_c = \frac{\int\limits_{\overline{\Omega}} (1 - H(\Phi)) w \, \mathrm{d}\Omega}{\int\limits_{\overline{\Omega}} (1 - H(\Phi)) \, \mathrm{d}\Omega}, \tag{8.6}$$

where $H$ is the Heaviside step function and the integration is over $\overline{\Omega}$, which is the part of the domain $\Omega$ that does not include the air above the free surface. Note that by definition

$$\Phi = \begin{cases} < 0, & \text{air}, \\ > 0, & \text{water}, \end{cases} \tag{8.7}$$

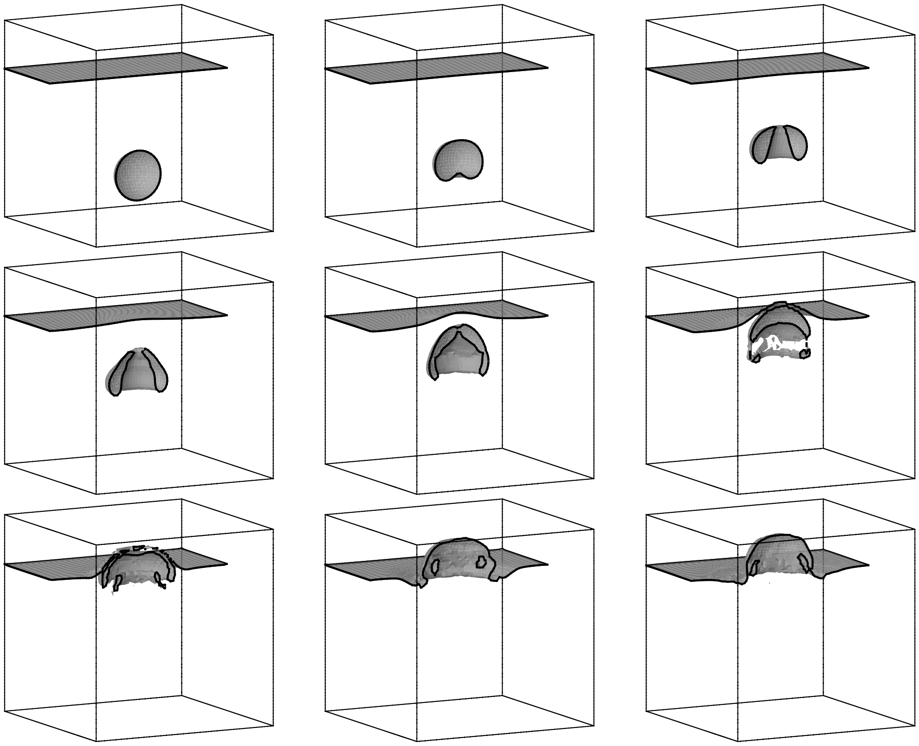


Figure 8.11: Rising bubble without surface tension; $64^3$ grid

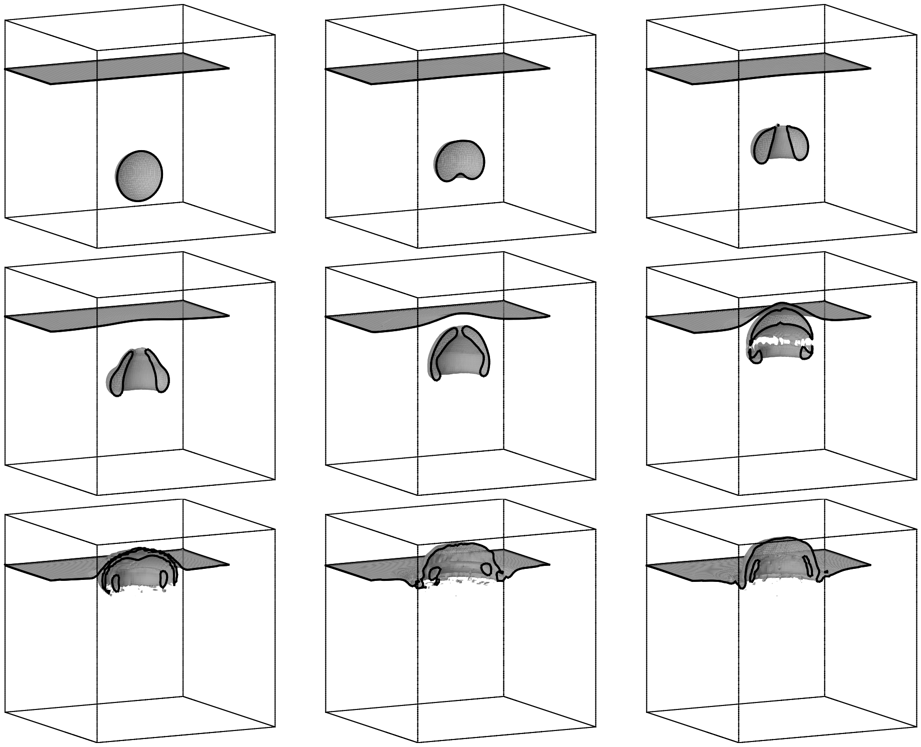


Figure 8.12: Rising bubble without surface tension; $96^3$ grid

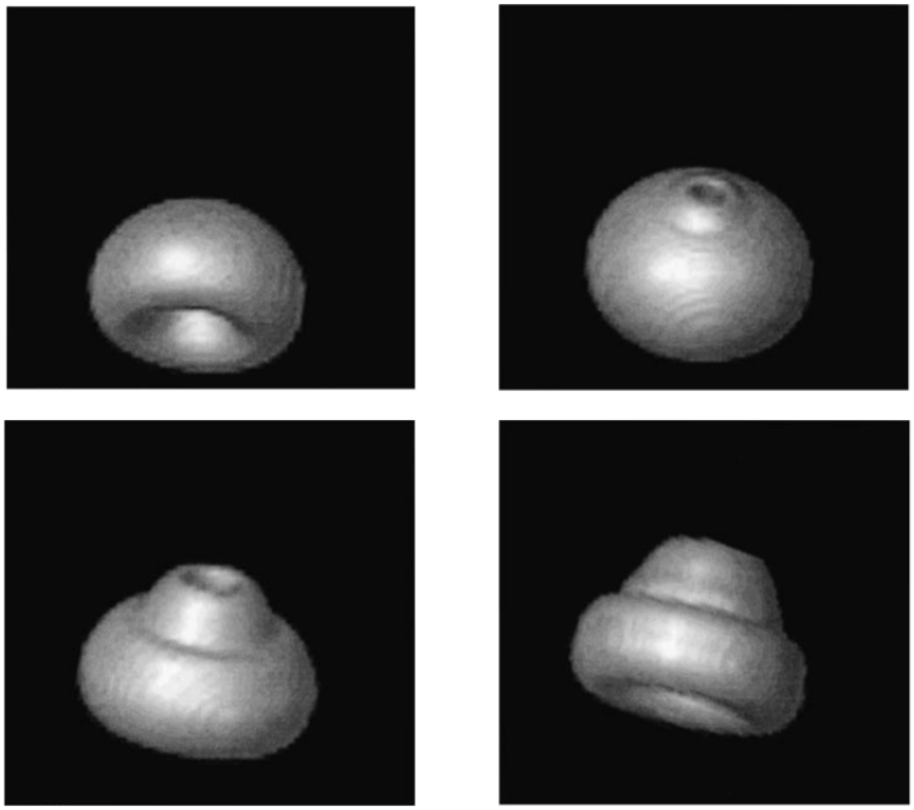
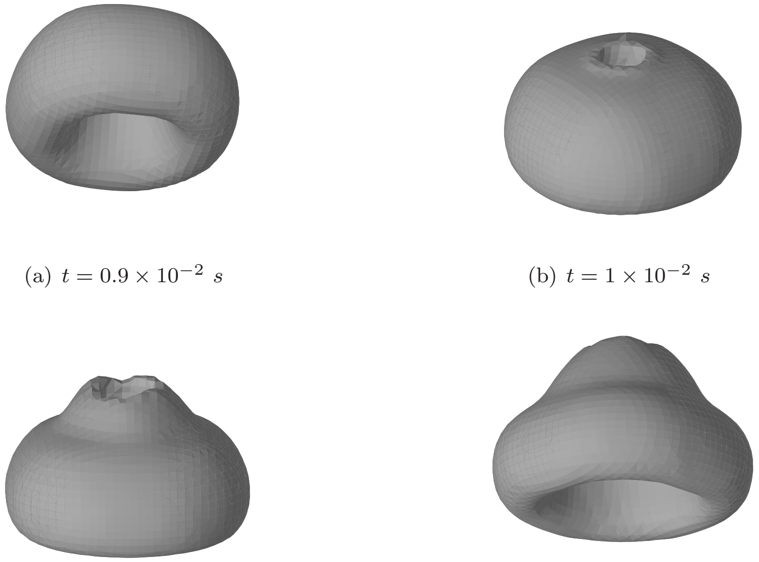


Figure 8.13: Rising bubble with low surface tension; courtesy of Sussman et al. [1]



(a) $t = 0.9 \times 10^{-2}\ s$ (b) $t = 1 \times 10^{-2}\ s$

(c) $t = 1.3 \times 10^{-2}\ s$ (d) $t = 1.5 \times 10^{-2}\ s$

Figure 8.14: Rising bubble without surface tension; $96^3$ grid; zoomed in

so that $1 - H(\Phi)$ is 1 in air and 0 in water. Using the Volume-of-Fluid function Eqn. (5.22) in computational cell $\Omega_k$

$$\Psi_k = \frac{1}{\mathrm{vol}(\Omega_k)} \int\limits_{\Omega_k} H(\Phi)\,\mathrm{d}\Omega,, \tag{8.8}$$

$w_c$ is approximated as

$$w_c = \frac{\sum\limits_k (1 - \Psi_k) w_k}{\sum\limits_k 1 - \Psi_k}, \tag{8.9}$$

where the summation is over all computational cells in $\overline{\Omega}$. The rise speed is in good agreement for both grids. The rise speed reaches a maximum just after the bubble starts moving. A local maximum is reached again before impact with the free surface at the top of the domain.

The shape of the bubble just before it breaks through the free surface is depicted in Fig. 8.16. The bubble takes the form of a spherical cap with radius $R_s = 0.0018\ m$. In [62] an expression for the rise speed of a spherical cap, based

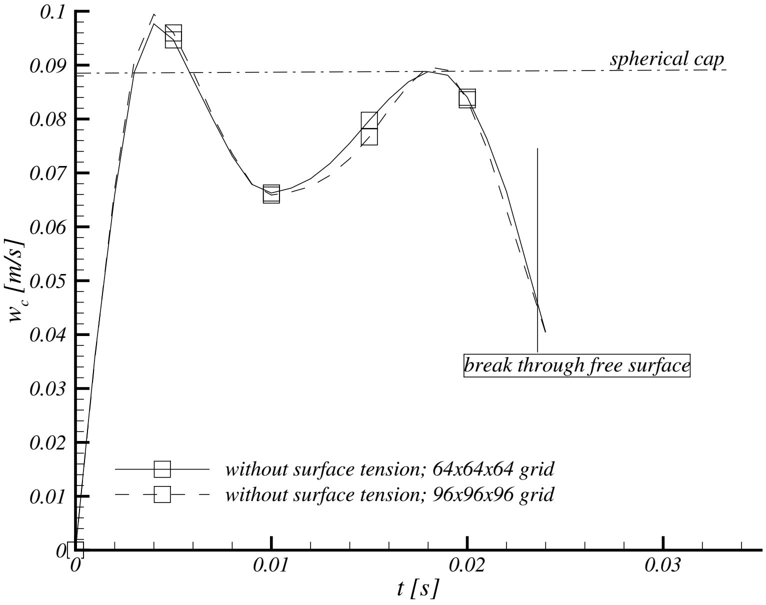


Figure 8.15: Rise speed of the bubble without surface tension; the dotted line is the analytic rise speed of the spherical cap

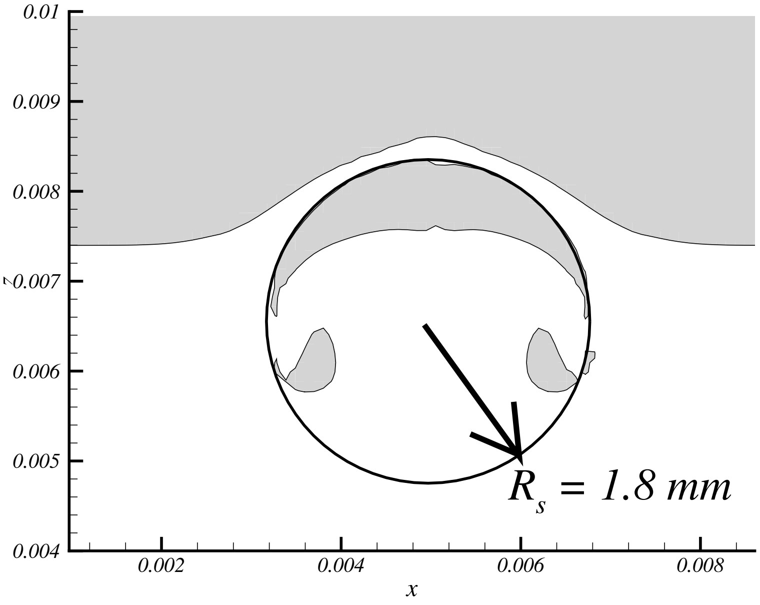


Figure 8.16: Bubble without surface tension just before it breaks through the free surface; symmetry plane $y = \frac{1}{2}L_y$; $R_s$ is the radius of the spherical cap.

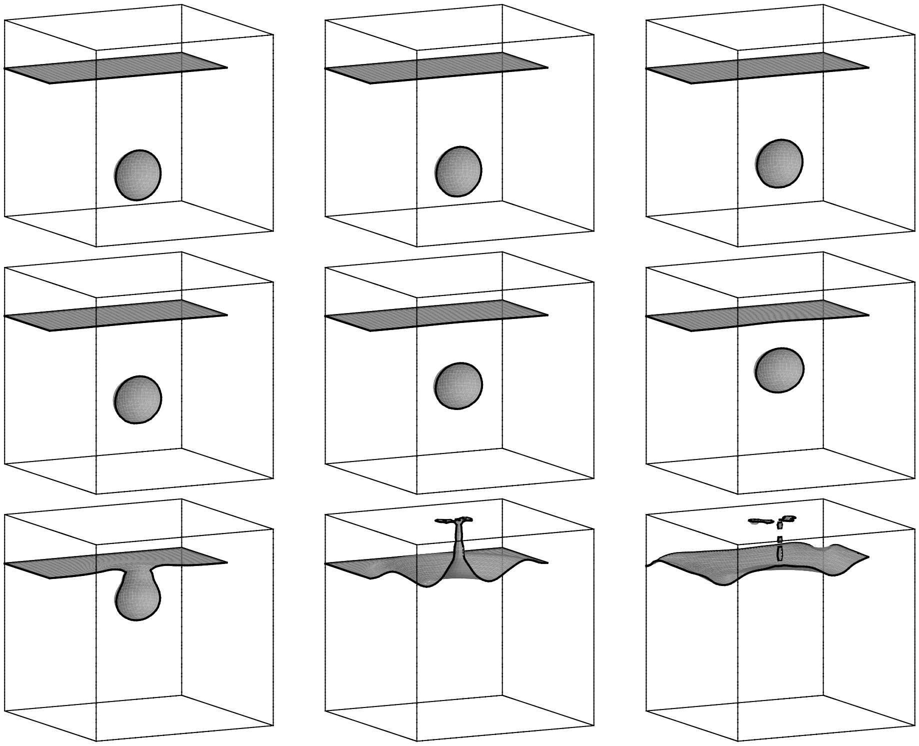


Figure 8.17: Rising bubble with surface tension; $64^3$ grid

on irrotational flow, is derived:

$$w_s = \frac{2}{3}\sqrt{gR_s} = 0.0885 m/s. \tag{8.10}$$

This speed is indicated in Fig. 8.15. The numerical results compare well with this speed until the bubble breaks through the free surface.

### 8.3.3 Rising air bubble in water with surface tension

In Figs. 8.17 and 8.18 results with surface tension are presented. The deformation of the bubbles is significantly reduced due to the surface tension effects. In Fig. 8.19 the rise speed $w_c$ of the bubble with surface tension is plotted for both grids. As a reference the rise speed of a bubble without surface tension (previous test case) is also included in this figure. The agreement between both grids is good. When the bubble is just released, the acceleration of the bubble with surface tension is lower than the bubble without surface tension. But instead of reaching a maximum rise speed, the bubble with surface tension continuous to accelerate further. Eventually higher speeds are reached for the bubble with surface tension than without surface tension. A peak is observed just before

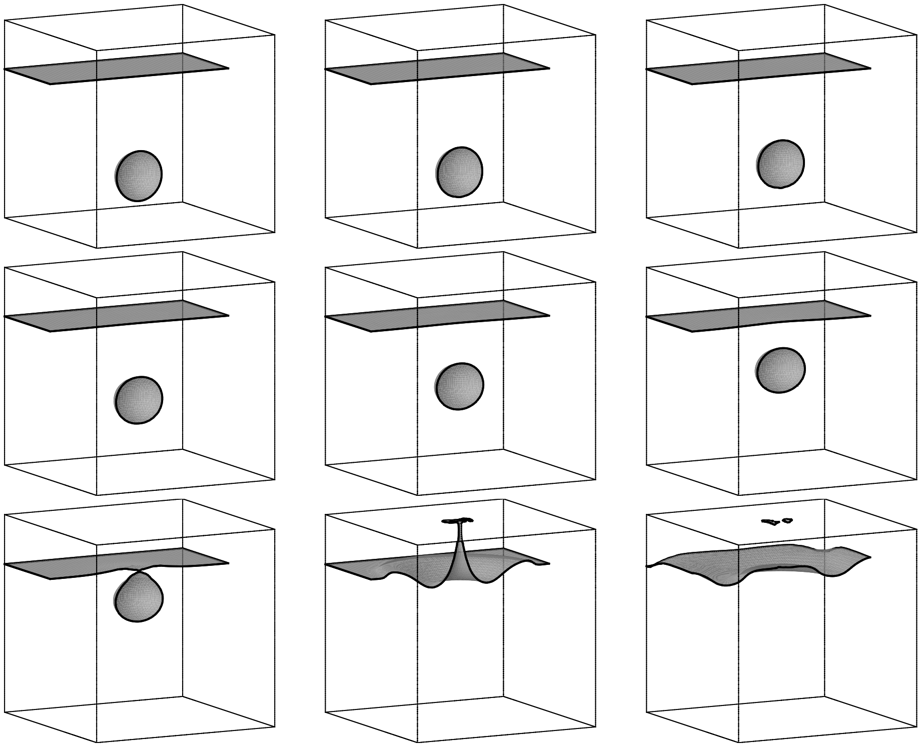


Figure 8.18: Rising bubble with surface tension; $96^3$ grid

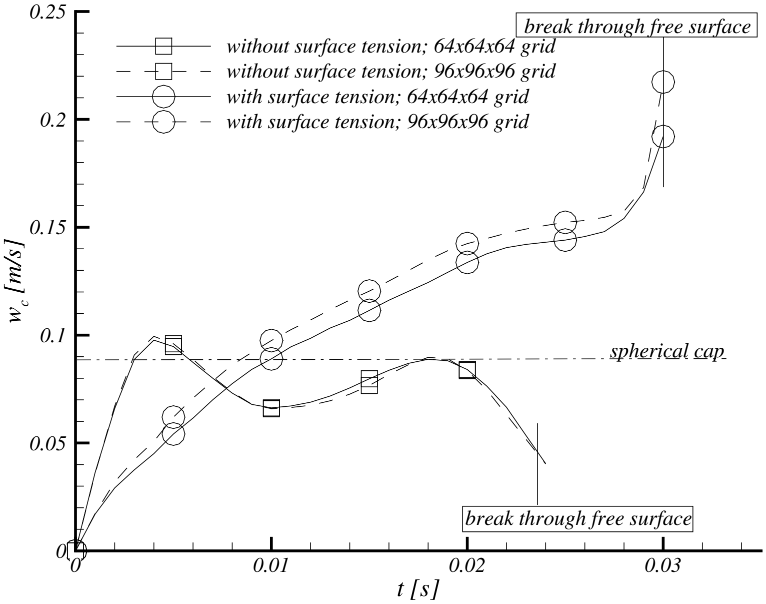


Figure 8.19: Rise speed of the bubble with surface tension; marks are at equally spaced time intervals of 0.005 $s$.

collision with the free surface at the top of the domain. This indicates that the bubble is sucked towards the free-surface before merging with it.

### 8.3.4 Falling water droplet in air without surface tension

In Fig. 8.20 the set up of a falling water droplet without surface tension is shown. The gravity and material constants are the same as for the bubble. The dimensions of the domain are also the same as for the rising-bubble test case. The droplet is released at half the height of the domain, i.e. $z_0 = 0.005\ m$ and the free surface is initially located at $1/4^{th}$ height: $z_1 = 0.0025\ m$. The radius of the droplet is $R = 0.00125\ m$. Results are shown in Figs. 8.21 and 8.22. The snapshots are taken at intervals of 0.01 $sec$. The droplet accelerates after it is released and hits the free surface. A jet appears after collision that reaches up to the ceiling of the domain.

In Fig. 8.23 the fall speed $w_c$ of the droplet before impact with the free surface is plotted as function of time $t$. For the fall speed $w_c$ the velocity of the center of gravity of the droplet is taken, which is computed in a similar fashion

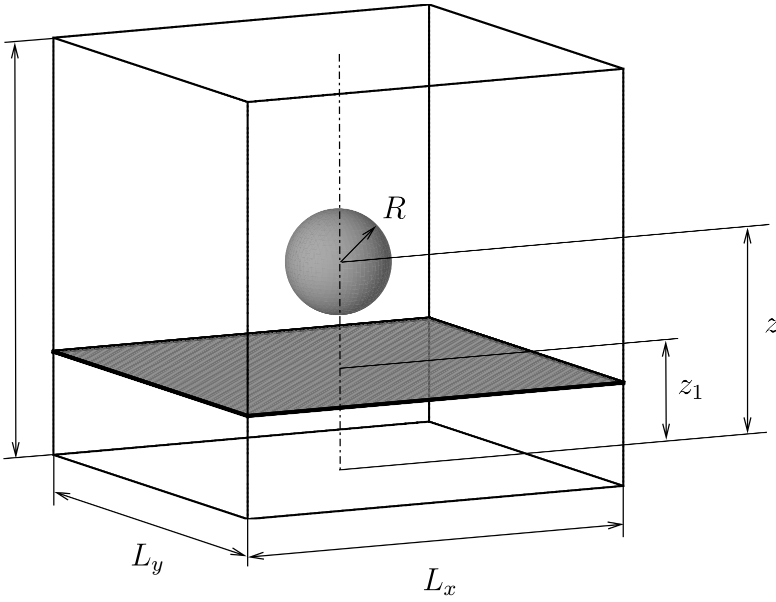


Figure 8.20: Set-up of the falling-droplet without surface tension test case

as the rise speed of a bubble; Eqn. (8.9) becomes

$$w_c = \frac{\sum\limits_k \Psi_k \mathbf{x}_k}{\sum\limits_k \Psi_k}. \tag{8.11}$$

Good agreement of the fall speed with the free-fall velocity $w_c = -gt$ is observed.

### 8.3.5 Falling water droplet in air with surface tension

The release height of a droplet with surface tension is increased to $z_0 = 0.0075\,m$. The fall height of the droplet without surface tension is smaller because otherwise the droplet becomes too deformed when it hits the free surface. The deformation of the droplet with surface tension is much less and allows a higher fall height. In Figs. 8.24 and 8.25 results are presented. The droplet indeed does not change shape before it collides with the free surface. Although the release height of the droplet is larger than without surface tension, the jet after collision is considerably smaller. This is the effect of surface tension that tries to minimize the interfacial area. In Fig. 8.26 the falling speed is compared with the free fall velocity $w_c = -gt$. The computed fall speeds for both grids are in agreement and larger than the free-fall velocity. Since the droplet without surface tension accelerated with $\frac{\mathrm{d}w_c}{\mathrm{d}t} = -g$ (see previous test case) the acceleration

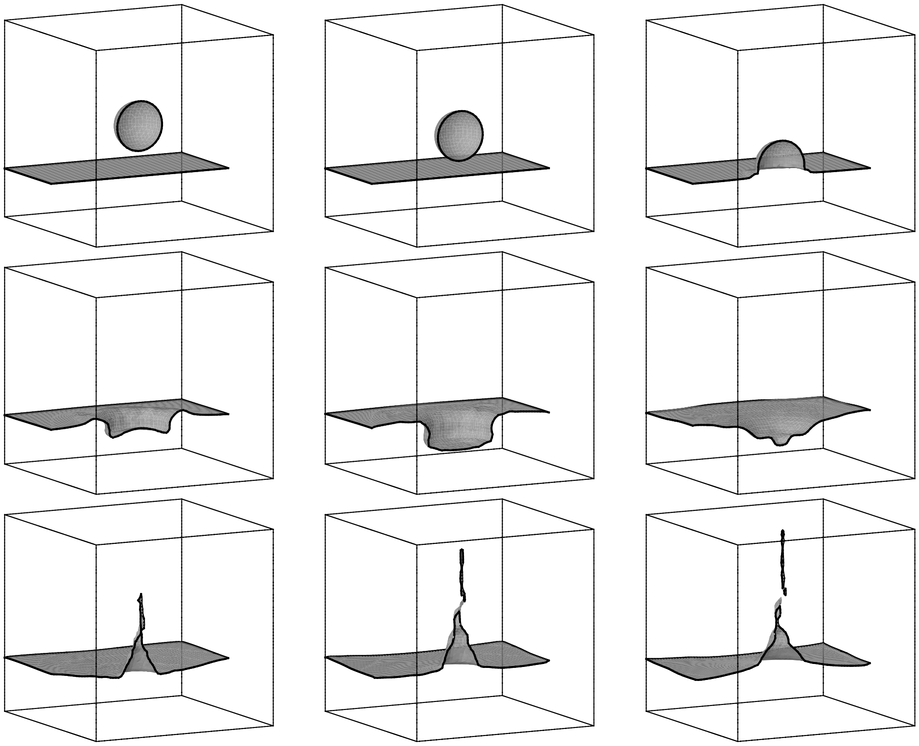


Figure 8.21: Falling droplet without surface tension; $64^3$ grid

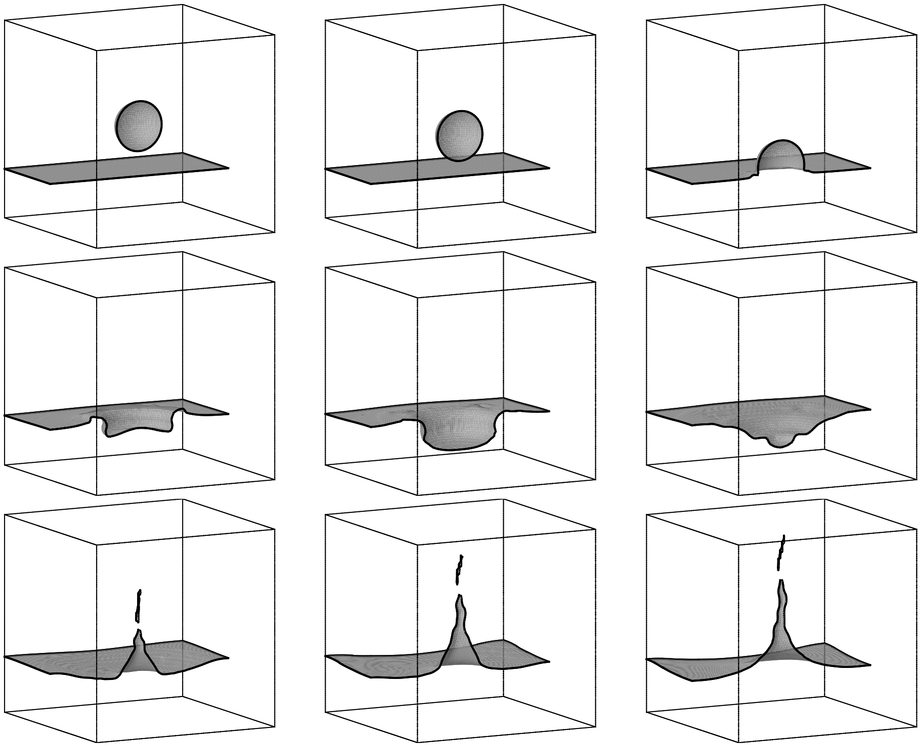


Figure 8.22: Falling droplet without surface tension; $96^3$ grid

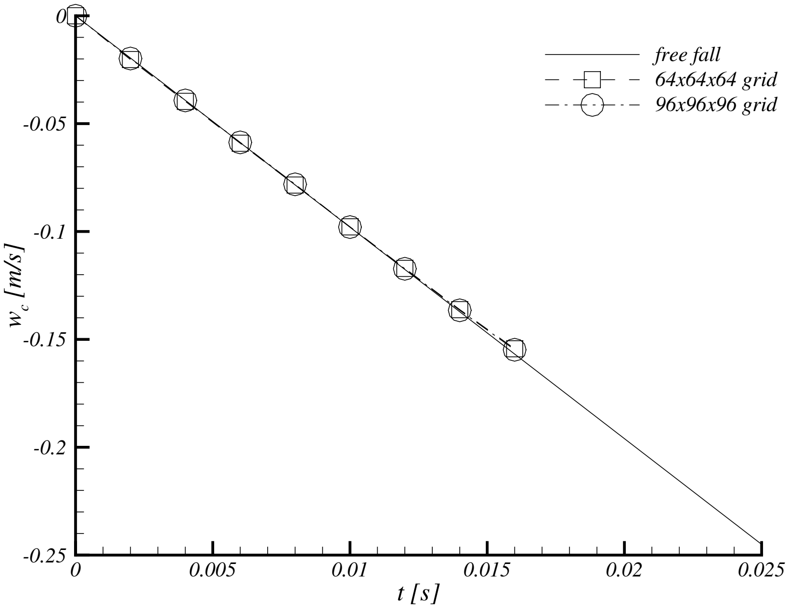


Figure 8.23: Fall speed of the droplet without surface tension compared with the free-fall velocity

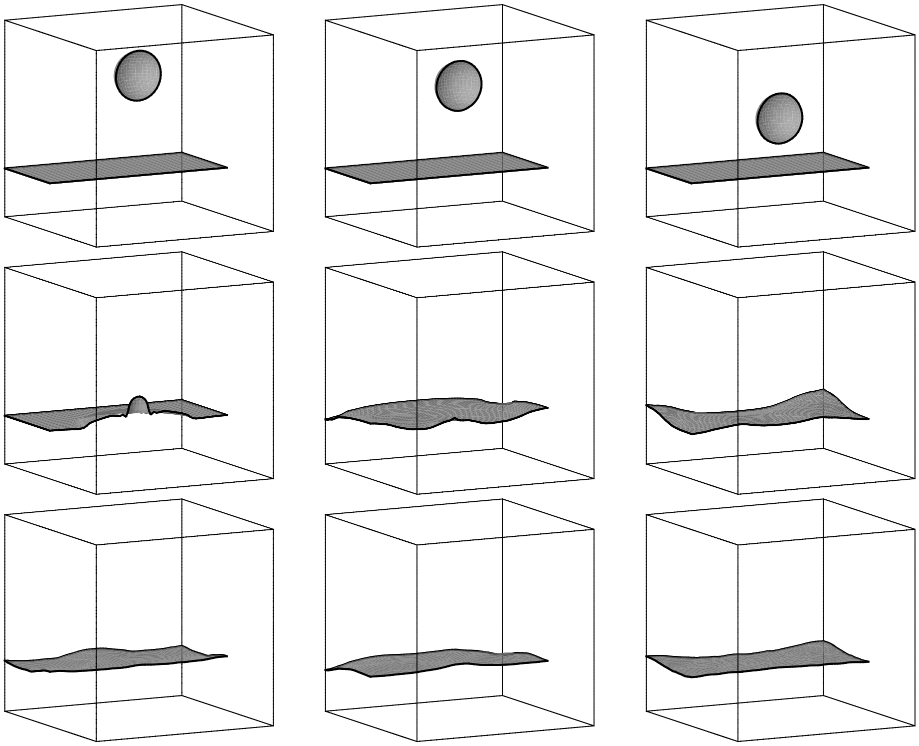


Figure 8.24: Falling droplet with surface tension; $64^3$ grid

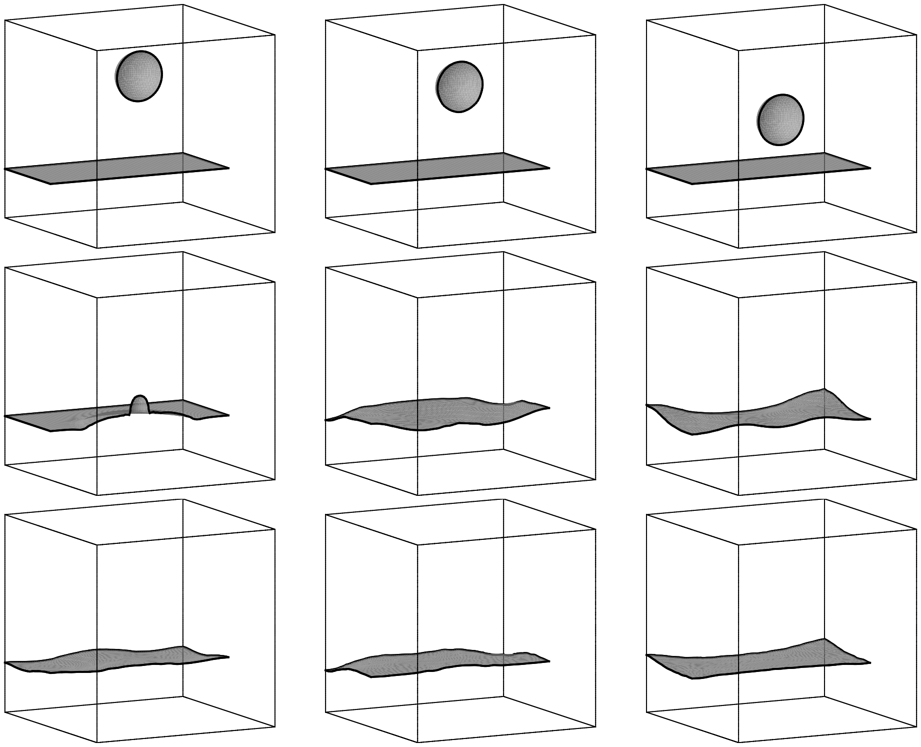


Figure 8.25: Falling droplet with surface tension; $96^3$ grid

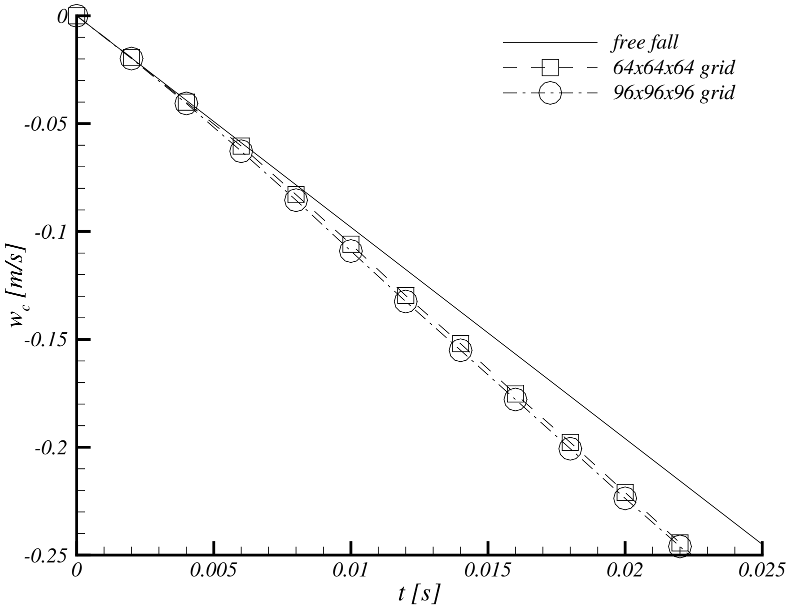


Figure 8.26: Fall speed of the droplet with surface tension compared with the free-fall velocity

with surface tension is higher than without surface tension. This is believed to be caused by the CSF approach (see Sections 2.6 end 4.3). The total force acting on a bubble or droplet due to surface tension should vanish, since the surface tension forces are internal forces and the interface is closed. However with the CSF approach a non-conservative expression is obtained (see Eqns. (2.22) and (4.6)). This means that numerically the forces do not necessarily add up to zero any longer. Inspection of the numerical data shows that

$$\sum_k \frac{1}{\rho}\mathbf{f}_s \approx -10^{-8}. \tag{8.12}$$

The acceleration of the center of gravity of the droplet will therefore be

$$g_z + \frac{1}{\frac{4}{3}\pi R^3}\sum_k \frac{1}{\rho}\mathbf{f}_s \approx -11 m/s^2, \tag{8.13}$$

where $\frac{4}{3}\pi R^3$ is the volume of the droplet. Note that $g_z$ is the $z$-component of the gravity vector $\gg= (g_x, g_y, g_z)^t$. This is in agreement with the results in Fig. 8.26, where the droplet indeed accelerates with approximately $-11$ $m/s^2$.

### 8.3.6 Comparison of rising bubble with experimental data

In [3] bubbly flows are simulated with a front tracking and a Volume-of-Fluid method. The rise speed is compared with experimental observations. Assume that '1' corresponds to the heavier fluid outside the bubble and '0' to the light fluid inside the bubble. The flow is characterized by the following dimensionless parameters:

$$\begin{aligned} \text{Eö} &= \frac{g(\rho_1-\rho_0)(2R)^2}{\sigma} \\ \text{Mo} &= \frac{g\mu_1^4(\rho_1-\rho_0)}{\rho_1^2\sigma^3}. \end{aligned} \tag{8.14}$$

where Eö is the Eötvos number and Mo is the Morton number. The final rise speed $w_c$ is then measured in the Reynolds number Re:

$$\text{Re} = \frac{\rho_l w_c 2R}{\mu_1}. \tag{8.15}$$

In [3] the following parameters are chosen:

$$\begin{aligned} \text{Eö} &= 1.0 \\ \log(\text{Mo}) &= -3.8. \end{aligned} \tag{8.16}$$

This corresponds to:

$$\begin{aligned} g &= 9.8\ \frac{m}{s^2}, & \sigma &= 0.0792\ \frac{kg}{s^2}, \\ \rho_0 &= 1.293\ \frac{kg}{m^3}, & \rho_1 &= 262\ \frac{kg}{m^3}, \\ \mu_0 &= 1.71\times 10^{-6}\ \frac{kg}{ms}, & \mu_1 &= 5.50\times 10^{-2}\ \frac{kg}{ms}. \end{aligned}$$

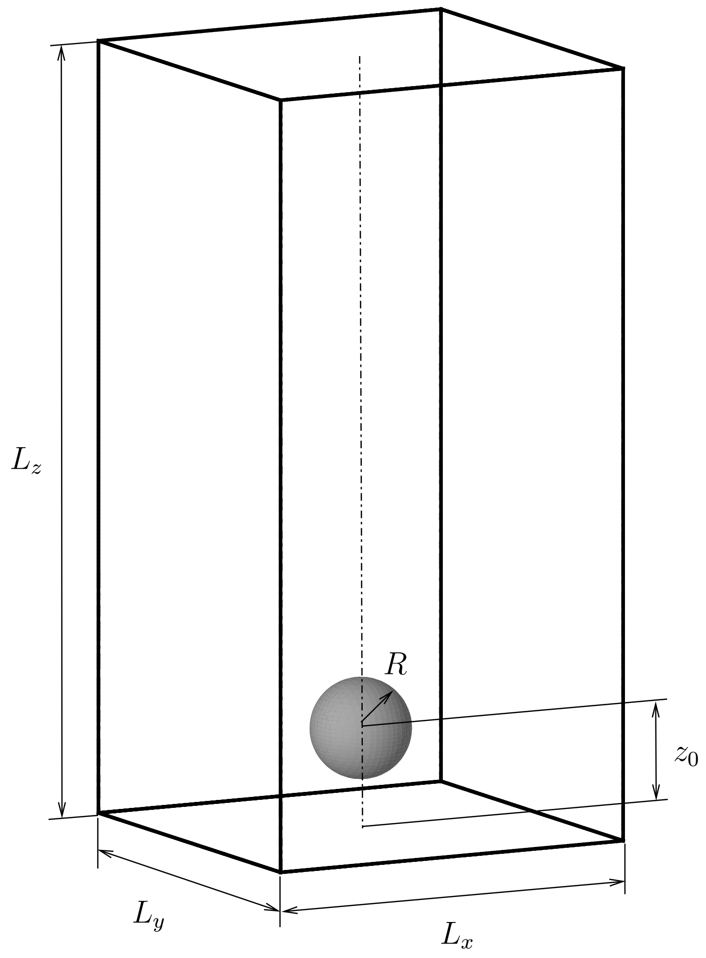


Figure 8.27: Set-up of the rising bubble test case

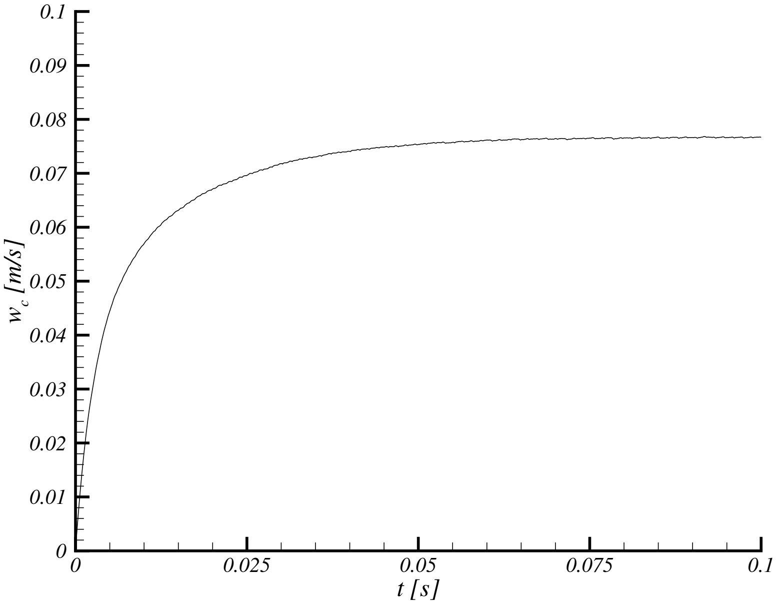


Figure 8.28: Rise speed of the bubble; Eö = 1.0; $\log(M) = -3.8$

The bubble radius is $R = 0.0013\,m$. In this work, the dimensions of the domain are: $L_x = 0.01\ m$, $L_y = L_x$, $L_z = 2L_x$, see Fig. 8.27. The initial distance from the bubble to the floor of the domain is: $z_0 = 2R$. The grid size is $64 \times 64 \times 192$. The computed rise speed of the bubble $w_c$ is presented in Fig. 8.28. The rise speeds converges monotonically to a constant value.

When the flow around the bubble is fully developed, the drag force $F_d$ is balanced by the buoyancy force, i.e.:

$$F_d = \frac{4}{3}\pi R^3 g(\rho_1 - \rho_0). \tag{8.17}$$

Experimental data is usually fitted with the drag force drag coefficient $c_d$

$$c_d = \frac{F_d}{\frac{1}{2}\rho_1 w_c^2 \pi R^2}. \tag{8.18}$$

In [3] three different experimental correlations are used to compare with the

| | | | Re |
|---|---|---|---|
| Gunsing | [3] | VOF | 8.2 |
| Gunsing | [3] | front tracking | 6.4 |
| experimental | (I) | Eqn. (8.19) | 5.2 |
| experimental | (II) | Eqn. (8.20) | 5.3 |
| experimental | (III) | Eqn. (8.21) | 5.8 |
| this work | | | 4.6 |

Table 8.3: Reynolds numbers of the rising bubble compared with the numerical results from Gunsing [3] and experimental data; Eö = 1.0; $\log(M) = -3.8$

numerical results. These are:

$$\begin{aligned} (I)\ c_d &= 14.9\mathrm{Re}^{-0.78}, && (8.19)\\ (II)\ c_d &= \frac{3.05(783\gamma^2+2142\gamma+1080)}{(60+29\gamma)(4+3\gamma)}\mathrm{Re}^{-0.74}, && (8.20)\\ (III)\ c_d &= \max(\min(\frac{A}{\mathrm{Re}_b(1+0.15\mathrm{Re}_b^{0.6882})}, \frac{3A}{\mathrm{Re}_b}), \frac{8}{3}\frac{\mathrm{Eö}}{\mathrm{Eö}+4}), && (8.21) \end{aligned}$$

where $\gamma = \frac{\rho_0}{\rho_1}$. For more detail on these experimental closures see [3]. The resulting Reynolds numbers are shown in Table 8.3. There is agreement in the order of magnitude of the rise speed. When compared to the experimental data, the results of Gunsing shows Reynolds number that is about 40% larger for the VOF method and 10% for the front tracking method. Our method under-predicts the rise speed with about 10%.

### 8.3.7 Coalescence of two aligned rising bubbles

The coalescence of two rising bubbles is studied by De Sousa et al. in [2]. The initial conditions are presented in Fig. 8.29. Two bubbles are initially at rest and aligned on the center line of the computational domain. The dimensions of the domain are: $L_x = 0.01\ m$, $L_y = 0.01\ m$, $L_z = 0.02\ m$. The radii of the bubbles are $R = 0.0013\ m$. The distance from the bottom bubble to the floor of the domain is: $z_0 = 0.0025\ m$. The distance between the bubbles is $\Delta z = 0.0055\ m$. Note that this makes the gap between the bubbles $\Delta z - 2R = 0.0004\ m$ (as prescribed by De Sousa et al. [2]). The gravity and material constants are:

$$\begin{aligned} g &= 9.8\,\frac{m}{s^2}, & \sigma &= 5.8\times10^{-4}\,\frac{kg}{s^2},\\ \rho_0 &= 440\,\frac{kg}{m^3}, & \rho_1 &= 880\,\frac{kg}{m^3},\\ \mu_0 &= 0.00625\,\frac{kg}{ms}, & \mu_1 &= 0.00125\,\frac{kg}{ms}, \end{aligned}$$

where subscripts $_0$ and $_1$ indicate the lighter and heavier fluid respectively. The coalescence of two rising bubbles is illustrated in Fig. 8.30. Snapshots are taken

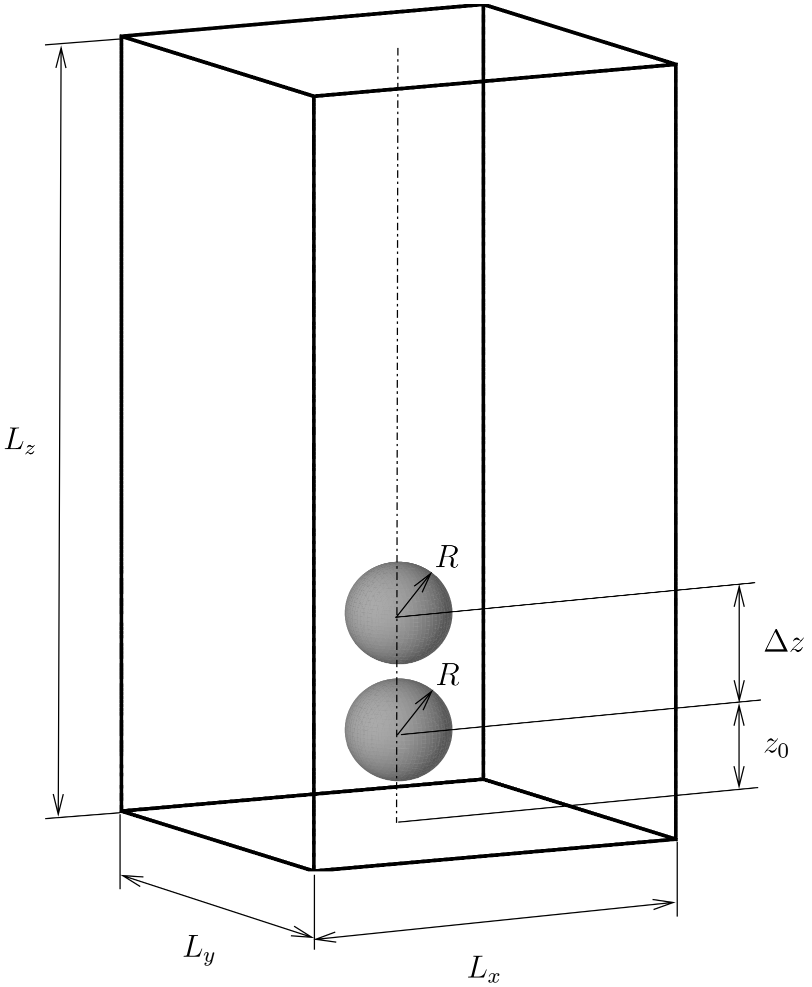
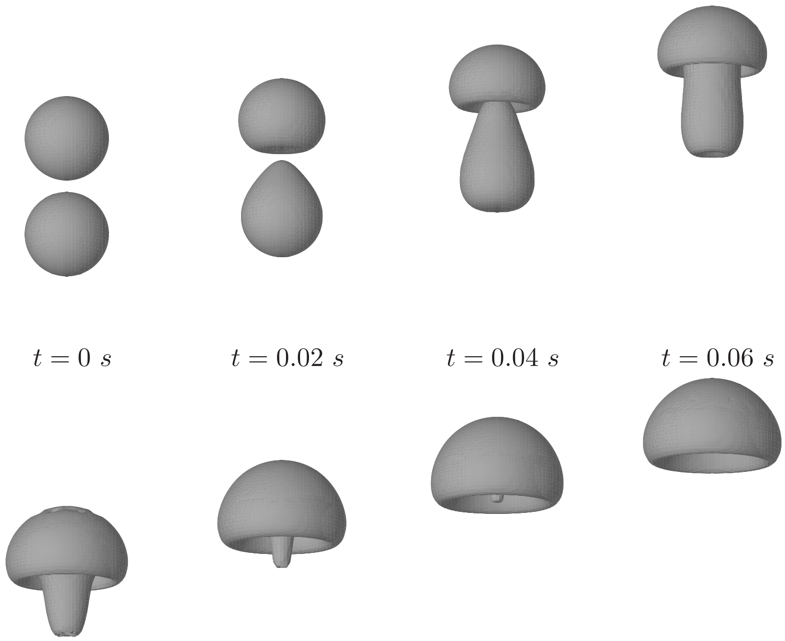


Figure 8.29: Set-up of the two aligned rising-bubbles test case



$t = 0\ s$ $t = 0.02\ s$ $t = 0.04\ s$ $t = 0.06\ s$

$t = 0.08\ s$ $t = 0.1\ s$ $t = 0.12\ s$ $t = 0.14\ s$

Figure 8.30: Coalescence of two aligned rising bubbles; $96 \times 96 \times 192$ grid

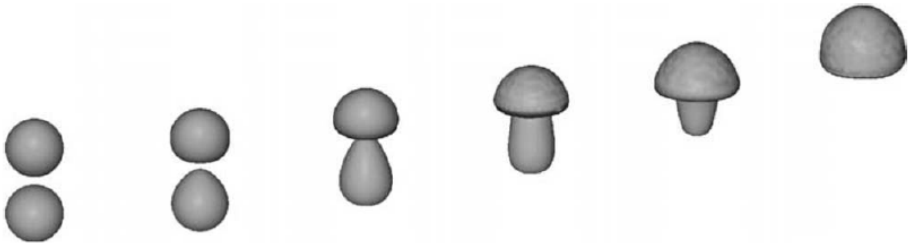


Figure 8.31: Coalescence of two aligned rising bubbles; courtesy of De Sousa et al. [2]

at equally spaced time intervals of 0.02 $s$, starting from the initial condition. Good qualitative agreement is found with the results presented by De Sousa et al. [2] (see Fig. 8.31). The snapshots presented in [2] are taken at intervals of 0.03 $s$, which means that the bubbles rise speed in this research is approximately 35% larger than in [2]. Note that velocities predicted by our model have been validated in the previous sections.

The vertical velocity of the center of gravity of the two bubbles is plotted in Fig. 8.32. Note that this velocity is computed as explained in Section 8.3.2. The rise speed increases to a maximum and converges to a smaller steady speed. In Section 8.3.2 a rising bubble without surface tension was studied. The rise speed reached a maximum just after the bubble was released. This also occurs here. This might be explained by the fact that here the surface tension coefficient is relatively small compared to the surface tension in the air-water test cases.

### 8.3.8 Coalescence of two misaligned rising bubbles

The bubbles are initially misaligned to study asymmetric merging, see Fig. 8.33. The set-up is the same as for the aligned rising bubbles, except for a horizontal distance $\Delta x$ between the bubbles. This distance is equal to the bubble radius, i.e. $\Delta x = R$. The results are plotted in Fig. 8.34. It can be seen that the bottom bubble is deformed considerably and sucked upward towards the top bubble. A thin tail is observed after merging. The tail becomes smaller, but never completely disappears. These simulations are also performed by De Sousa et al. [2], see Fig. 8.35. The qualitative agreement is good, albeit that the rise speed is different. This was also the case for the aligned bubbles. Furthermore, the tail after merging of the two bubbles suddenly disappears in [2], whereas according to Fig. 8.34 the tail does not disappear. The tail remains but becomes thinner as time passes. The interface does also seems to be more irregular in [2], despite of the so-called ‘undulation removal procedure’ carried out by De Sousa et al. [2] to smooth the interface. Note that in this work the interface is not smoothed. Still a smooth interface is observed.

The vertical velocity of the center of gravity of the two bubbles is plotted

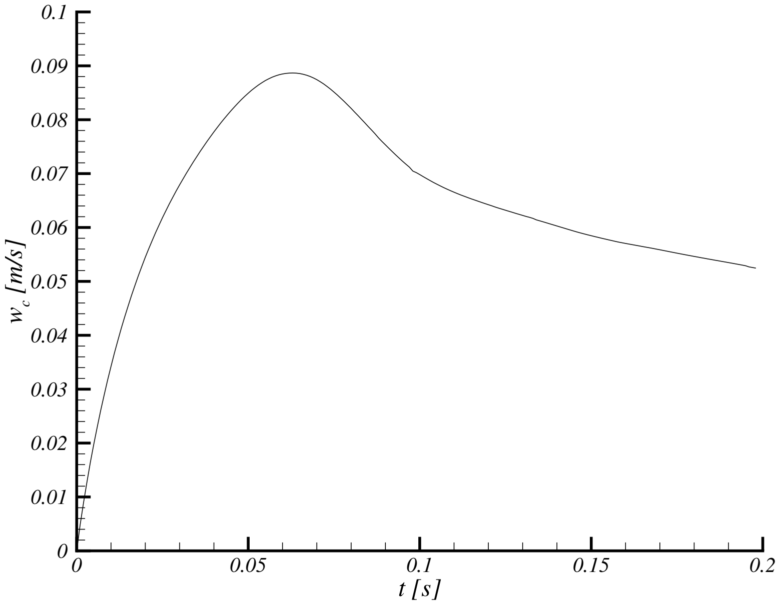


Figure 8.32: Rise speed of the aligned bubbles; 96 × 96 × 192 grid

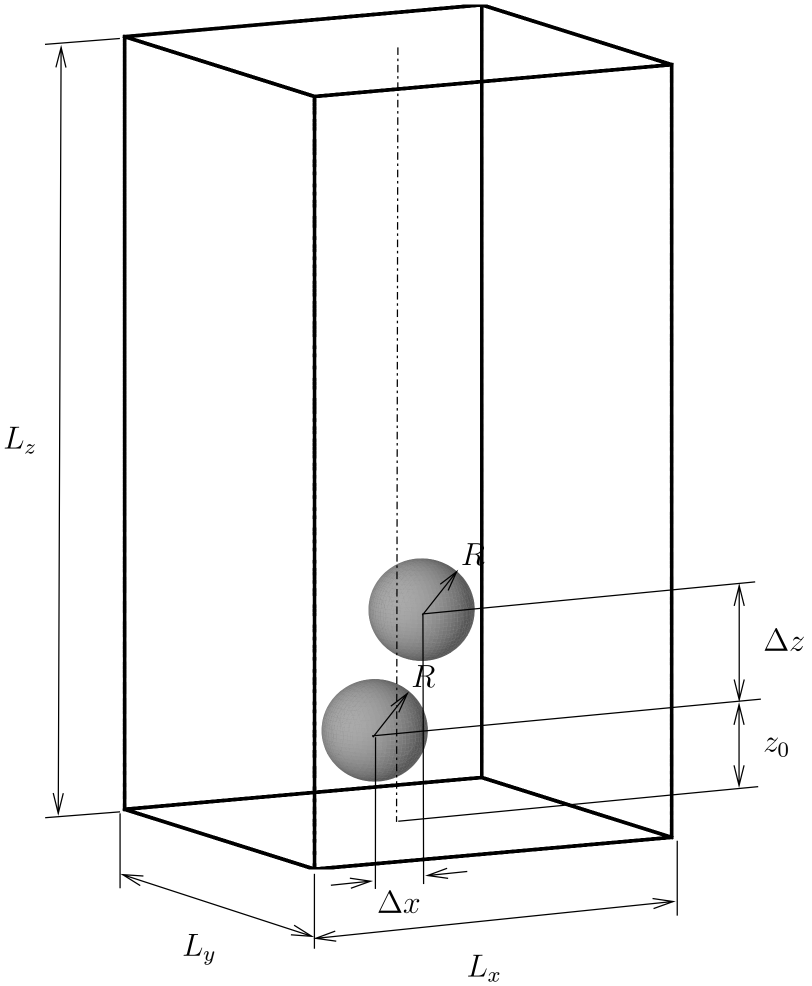
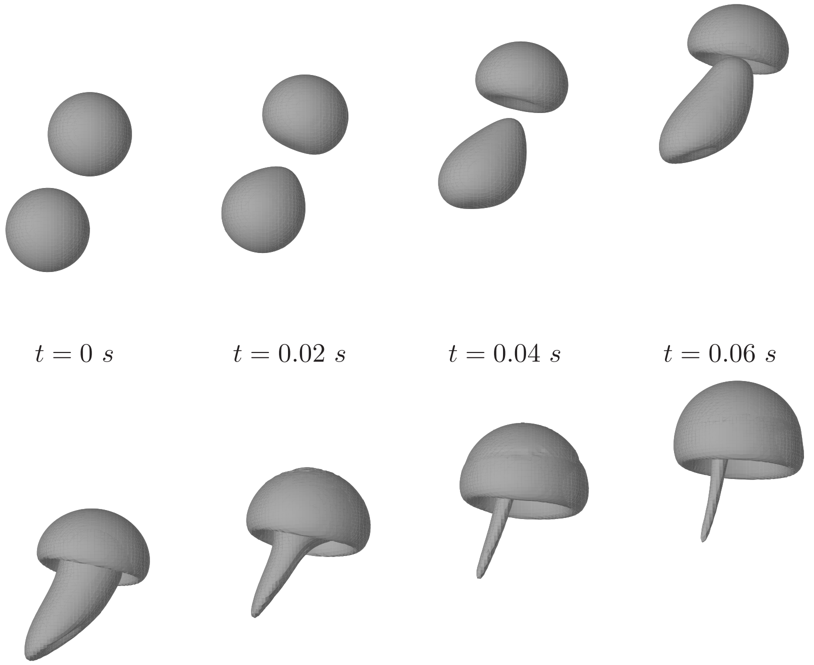


Figure 8.33: Set-up of the two misaligned rising bubbles



$t = 0\ s$ $t = 0.02\ s$ $t = 0.04\ s$ $t = 0.06\ s$

$t = 0.08\ s$ $t = 0.1\ s$ $t = 0.12\ s$ $t = 0.14\ s$

Figure 8.34: Coalescence of two misaligned rising bubbles; 96 × 96 × 192 grid

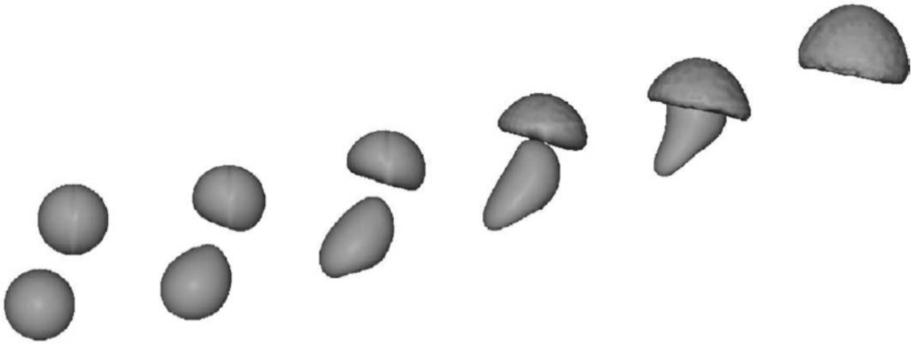


Figure 8.35: Coalescence of two misaligned rising bubbles; courtesy of De Sousa et al. [2]

in Fig. 8.36. As a reference, the rise speed of the aligned bubble test case is also included in this figure. The behavior is the same as in the previous case. The maximum rise speed reached is lower. The speed converges to the speed of the misaligned bubbles. This is to be expected, as the shape of the merged, aligned bubble becomes similar to the shape of the merged, misaligned bubble, see Figs. 8.30 and 8.34. The location of the center of gravity in the $x-z$ symmetry plane is plotted in Fig. 8.37. This is computed in the same way as the rise speed $w_c$:

$$\mathbf{x}_c = \frac{\sum\limits_k (1-\Psi_k)\mathbf{x}_k}{\sum\limits_k (1-\Psi_k)}, \tag{8.22}$$

where the summation is over all computational cells $k$. The figure shows that the aligned bubbles move straight up. The misaligned bubbles deviate in the direction of the top bubble at first, but eventually move straight up. The marks are placed at equally spaced time intervals of 0.02 for comparison with Figs. 8.30 and 8.34. The rise speed is clearly at maximum halfway the domain, which was also concluded from Fig. 8.36.

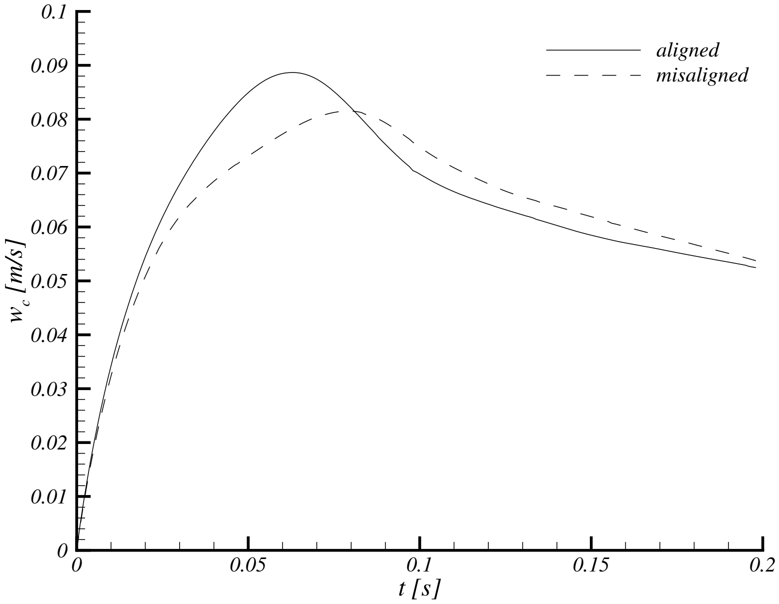


Figure 8.36: Rise speed of the bubbles; 96 × 96 × 192 grid

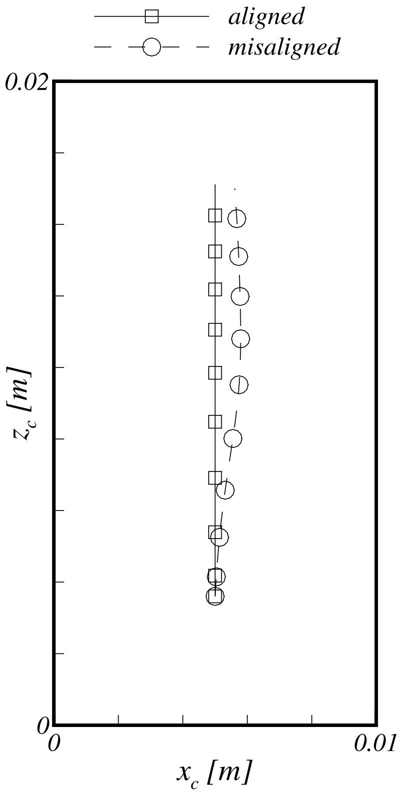


Figure 8.37: Location of the center of gravity $(x_c, z_c)$ of the rising bubbles in the $x - z$ symmetry plane; marks are at equally spaced time intervals of 0.02 $s$; $96 \times 96 \times 192$ grid

# Chapter 9

# Conclusions

The purpose of this work is to model bubbly flows. This has been achieved by using the Level-Set approach. It appears that the Level-Set method does not conserve the mass of each separate phase. Therefore a 'Mass Conserving Level-Set' method was developed. It uses a so-called Volume of Fluid function to enhance mass conservation. The Volume-of-Fluid function measures the fractional volume of fluid within a computational cell. A straightforward relationship was found between the Volume-of-Fluid function and the Level-Set function, which makes the method easy to implement and to generalize to three dimensions. In that sense this work combines the appealing features of the Level-Set method with the mass-conservation properties of Volume-of-Fluid methods.

The viscosity and surface tension are regularized in a small band around the interface. The width of this band depends on the gradients of the Level-Set function. Re-initialization of the Level-Set function is used to keep the Level-Set function close to a distance function; the norm of the gradient of a distance function is one. The re-initialization procedure, however, causes the interface to shift. A simple modification is applied to the original procedure to remedy this defect.

In the 'Mass Conserving Level-Set' approach, corrections are made to the Level-Set function after each time step in order to conserve mass. The corrections are made locally near the interface only. Under certain circumstances, this might cause problems for the computation of the curvature, which depends on first and second order derivatives of the Level-Set function. So-called parasitic currents can occur. Complicating matters, the corrections become larger when re-initialization is performed, resulting in larger parasitic currents. This is overcome by introducing a start criterion for re-initialization and by carefully smoothing the curvature. It is shown that parasitic currents are dramatically reduced due to these modifications.

The mass-conserving properties of the current approach were shown by advection tests in two and three dimensions. The applicability of the method was illustrated by application to air-water flows. Both droplets and bubbles are simulated with and without surface tension. Arbitrarily shaped interfaces were encountered when bubbles merge and droplets collide with free surfaces. The results compare well with numerical results available from literature.

## 9.1 Recommendations for future research

The focus of this work has been on the numerics of modeling bubbly flows. Although comparison is made with other numerical work, there is a need for experimental validation.

The 'Mass Conserving Level-Set' method can now be used to study the physics of bubbly flow. Some examples are the interaction between bubbles and the effect of external flow on the shape and path of bubbles. Such fundamental research might be used to validate higher-level two-fluid modeling of multiphase flow, as commonly used in industrial applications. This will require the simulation of a larger number of droplets and bubbles.

In this work a limited number of drops and bubbles have been considered. These computations were performed with grid sizes up to about 2 million points. Simulations with a larger number of droplets or bubbles will need larger grid sizes. To keep computing time within limits this will, in turn, require parallelization of the computer code. This is believed to be a straightforward task, except for the Krylov solver. Parallelization of the Krylov solver will need special attention.

# Bibliography

[1] M. Sussman, A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, and M.L. Welcome. An adaptive level set approach for incompressible two-phase flows. *Journal of Computational Physics*, 148:81–124, 1999.

[2] F.S. de Sousa, N. Mangiavacchi, L.G. Nonato, A. Castelo, M.F. Tomé, V.G. Ferreira, J.A. Cumanito, and S. McKee. A front-tracking/front-capturing method for the simulation of 3D multi-fluid flows with free surfaces. *Journal of Computational Physics*, 198:469–499, 2004.

[3] M. Gunsing. *Modelling bubbly flows using volume of fluid, front tracking and discrete bubble models.* PhD thesis, University of Twenty, The Netherlands, 2004.

[4] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan. A front-tracking method for the computation of multiphase flow. *Journal of Computational Physics*, 169:708–759, 2001.

[5] Z. Li and M.-C. Lai. The immersed interface method for the Navier-Stokes equations with singular forces. *Journal of Computational Physics*, 171:822–842, 2001.

[6] J.U. Brackbill, D.B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100:335–354, 1992.

[7] W.Shyy, H.S.Udaykumar, M.M. Rao, and R.W. Smith. *Computational Fluid Dynamics with Moving Boundaries.* Series in computational and physical processes in mechanics and thermal sciences. Taylor and Francis, Washington DC, 1996.

[8] S.O.Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100:25–37, 1992.

[9] M.-C. Lai and C.S. Peskin. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics*, 160:705–719, 2000.

[10] H.S. Udaykumar, R. Mittal, P. Rampunggoon, and A. Khanna. A sharp interface cartesian grid method for simulating flows with complex moving boundaries. *Journal of Computational Physics*, 174:345–380, 2001.

[11] T. Ye, W. Shyy, , and J.N. Chung. A fixed-grid, sharp-interface method for bubble dynamics and phase change. *Journal of Computational Physics*, 174:781–815, 2001.

[12] F.H. Harlow and J.E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surfaces. *Physics of Fluids*, 8:2182–2189, 1965.

[13] M. Sussman and E.G. Puckett. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, 162:301–337, 2000.

[14] M. Sussman. 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.

[15] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183:83–116, 2002.

[16] D. Enright, D. Nguyen, F. Gibou, and R. Fedkiw. Using the particle level set method and a second order accurate pressure boundary condition for free surface flows. In *Proceedings of the FEDSM03 3rd ASME/JSME Joint Fluids Engineering Conference*, 2003.

[17] S.T. Zalesak. Fully multidimensional flux-corrected transport algorithm for fluids. *Journal of Computational Physics*, 31:335–362, 1979.

[18] W.F. Noh and P. Woodward. SLIC (Simple Line Interface Calculations). In A.I. van de Vooren and P.J. Zandbergen, editors, *Lecture Notes in Physics, Vol. 59*, pages 330–340, New York, 1976. Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics, Springer.

[19] C.W. Hirt and B.D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.

[20] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti. Modelling merging and fragmentation in multiphase flows with SURFER. *Journal of Computational Physics*, 113:134–147, 1994.

[21] S. Popinet and S. Zaleski. A front-tracking algorithm for accurate representation of surface tension. *International Journal for Numerical Methods in Fluids*, 30:775–793, 1999.

[22] M. Renardy, Y. Renardy, and J. Li. Numerical simulation of moving contact line problems using a volume-of-fluid method. *Journal of Computational Physics*, 171:243–263, 2001.

[23] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics*, 152:423–456, 1999.

[24] W.J. Rider and D.B. Kothe. Reconstructing volume tracking. *Journal of Computational Physics*, 141:112–152, 1998.

[25] M. Rudman. Volume-tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids*, 24:671–691, 1997.

[26] M. Rudman. A volume-tracking methods for incompressible multifluid flow with large density variations. *International Journal for Numerical Methods in Fluids*, 28:357–378, 1998.

[27] Scardovelli R and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31:567–603, 1999.

[28] D.J.E. Harvie and D.F. Fletcher. A new volume of fluid advection algorithm: the stream scheme. *Journal of Computational Physics*, 162:1–32, 2000.

[29] D.J.E. Harvie and D.F. Fletcher. A new volume of fluid advection algorithm: the defined donating region scheme. *International Journal for Numerical Methods in Fluids*, 35:151–172, 2001.

[30] J. Gerrits. *Dynamics of Liquid-Filled Spacecraft, Numerical Simulations of Coupled Solid-Liquid Dynamics*. PhD thesis, University of Groningen, The Netherlands, 2001.

[31] J. Gerrits and A.E.P. Veldman. Dynamics of liquid-filled spacecraft. *Journal of Engineering Mathematics*, 45:21–38, 2003.

[32] E. Loots, B. Hillen, and A.E.P. Veldman. The role of hemodynamics in the development of the outflow tract of the heart. *Journal of Engineering Mathematics*, 45:91–104, 2003.

[33] K.M.T. Kleefsman, G. Fekken, A.E.P. Veldman, B. Iwanowski, and B. Buchner. A volume-of-fluid based simulation method for wave impact problems. *Journal of Computational Physics*, 206:363–393, 2005.

[34] J.P. Boris and D.L. Book. Flux-Corrected Transport. I. SHASTA, a fluid transport algorithm that works. *Journal of Computational Physics*, 11:38–69, 1973.

[35] T. Yabe, F. Xiao, and T. Utsumi. The constrained interpolation profile method for multiphase analysis. *Journal of Computational Physics*, 169:556–593, 2001.

[36] Y.C. Chang, T.Y. Hou, B. Merriman, and S. Osher. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics*, 124:449–464, 1996.

[37] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.

[38] A. Iafrati, A. Di Mascio, and E.F. Campana. A level set technique applied to unsteady free surface flows. *International Journal for Numerical Methods in Fluids*, 35:281–297, 2001.

[39] W. Mulder, S. Osher, and J.A. Sethian. Computing interface motion in compressible gas dynamics. *Journal of Computational Physics*, 100:209–228, 1992.

[40] J. Zhu and J. Sethian. Projection methods coupled to level set interface techniques. *Journal of Computational Physics*, 102:128–138, 1992.

[41] J.A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge monographs on applied and computational mathematics. Cambridge University Press, Cambridge, 1999.

[42] J.A. Sethian. Evolution, implementation, and application of level set and fast marching methods for advancing fronts. *Journal of Computational Physics*, 169:503–555, 2001.

[43] S. Osher and R.P. Fedkiw. Level set methods: An overview and some recent results. *Journal of Computational Physics*, 169:463–502, 2001.

[44] M. Sussman, E. Fatemi, P. Smereka, and S. Osher. An improved level set method for incompressible two-phase flows. *Computers and Fluids*, 27:663–680, 1998.

[45] M. Sussman and E. Fatemi. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM Journal on Scientific Computing*, 20(4):1165–1191, 1999.

[46] L.L. Zheng and H. Zhang. An adaptive level set method for moving-boundary problems: Application to droplet spreading and solidification. *Numerical Heat Transfer, Part B*, 37:437–454, 2000.

[47] Ch. Kaliakatsos and S. Tsangaris. Motion of deformable drops in pipes and channels using Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 34:609–626, 2000.

[48] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *Journal of Computational Physics*, 83:32–78, 1989.

[49] A. Harten, B. Enquist, S. Osher, and S.R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 71:231–303, 1987.

[50] A. Harten. ENO schemes with subcell resolution. *Journal of Computational Physics*, 83:148–184, 1989.

[51] H. Haj-Hariri and Q. Shi. Thermocapillary motion of deformable drops at finite Reynolds and Marangoni numbers. *Physics of Fluids*, 9:845–855, 1997.

[52] J. Strain. A fast semi-Lagrangian contouring method for moving interfaces. *Journal of Computational Physics*, 170:373–394, 2001.

[53] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level set method. *Journal of Computational Physics*, 155:410–438, 1999.

[54] S. Shin and D. Juric. Modeling of three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity. *Journal of Computational Physics*, 180:427–470, 2002.

[55] S. Shin, S.I. Abdel-Khalik, V. Daru, and D. Juric. Accurate representation of surface tension using the level contour reconstruction method. *Journal of Computational Physics*, 203:493–516, 2005.

[56] H.D. Ceniceros and A.M. Roma. A multi-phase flow method with a fast, geometry-based fluid indicator. *Journal of Computational Physics*, 205:391–400, 2005.

[57] J.J.I.M. van Kan. A second-order accurate pressure correction method for viscous incompressible flow. *SIAM J. Sci. Stat. Comp.*, 7:870–891, 1986.

[58] M. Kang, R.P. Fedkiw, and X.-D. Liu. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, pages 323–360, 2000.

[59] X.-D. Liu, R.P. Fedkiw, and M. Kang. A boundary condition capturing method for Poisson's equation on irregular domains. *Journal of Computational Physics*, 160:151–178, 2000.

[60] S.P. van der Pijl, A. Segal, C. Vuik, and P. Wesseling. A mass-conserving Level-Set method for modelling of multi-phase flows. *International Journal for Numerical Methods in Fluids*, 47:339–361, 2005.

[61] E.R.A. Coyajee, M. Herrmann, and B.J. Boersma. Simulation of dispersed two-phase flow with a coupled Volume-of-Fluid/Level-Set method. In *Proceedings of the Summer Program 2004*. Center of Turblence Research, 2004.

[62] G.K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, UK, 1967.

# Curriculum Vitae

- 2001-present: Delft University of Technology, PhD student in Numerical Analysis, supported by the Netherlands Organization for Scientific Research (NWO), subject: *Computation of bubbly flows with a mass-conserving Level-Set method*, primary advisor: Prof.dr.ir. P. Wesseling, Department of Applied Mathematical Analysis, Faculty of Electrical Engineering, Mathematics and Computer Science
- 1999-2001: National Aerospace Laboratory, Amsterdam, research in the field of Computational Fluid Dynamics and aerodynamics
- 1999: Twente University, Enschede, research in the field of Computational Fluid Dynamics and aerodynamics
- 1998-1999: Delft University of Technology, research in the field of Computational Fluid Dynamics and aerodynamics
- 1995: National Aerospace Laboratory, Marknesse, internship in the field of aerodynamics
- 1995: Stichting Vliegschool Zestienhoven, Rotterdam, flying courses
- 1992-1998: Delft University of Technology, Delft, MSc degree in Aerospace Engineering, cum laude
- 1986-1992: Albanianae, Alphen aan den Rijn, VWO
- Born on 05-09-1974 in Ter Aar, The Netherlands

# List of Publications

- S.P. van der Pijl, A. Segal, and C. Vuik. A mass-conserving Level-Set method for modelling of multi-phase flows. *International Journal for Numerical Methods in Fluids*, 47:339–361, 2005.
- S.P. van der Pijl, A. Segal, and C. Vuik. Modelling of three-dimensional bubbly flows with a mass-conserving Level-Set method. In: K.J. Bathe, editor, *2005 MIT Conference Proceedings, Computational Fluid and Solid Mechanics*, pages 905-909, Elsevier, 2005
- S.P. van der Pijl, A. Segal, and C. Vuik. Modelling of three-dimensional bubbly flows with a mass-conserving Level-Set method. In P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, J. Périaux, and D. Knörzer, editors, *Proceedings European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2004, Jyväskylä, Finland, July 24-28, 2004*, Jyväskylä, 2004.
- S.P. van der Pijl, A. Segal, and C. Vuik. Modelling of multi-phase flows with a Level-Set method. In M. Feistauer, V. Dolejší, P. Knobloch, and K. Najzar, editors, *Numerical Mathematics and Advanced Applications*, pages 698–707, Springer, 2004.