Doctoral theses at NTNU, 2012:164

Claudio Walker

Numerical Methods for Two-Phase Flow with Contact Lines

Claudio Walker

Doctoral Thesis

ISBN 978-82-471-3617-1
ISBN 978-82-471-3619-5
ISSN 1503-8181

Doctoral theses at NTNU, 2012:164

NTNU
Norwegian University of Science and Technology
Thesis for the degree of Philosophiae Doctor
Faculty of Engineering Science & Technology
Department of Energy and Process Engineering

NTNU – Trondheim
Norwegian University of
Science and Technology

NTNU

NTNU – Trondheim
Norwegian University of
Science and Technology

Claudio Walker

Numerical Methods for Two-Phase Flow with Contact Lines

Thesis for the degree of Philosophiae Doctor

Trondheim, July 2012

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

NTNU – Trondheim
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

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

ISBN 978-82-471-3617-1
ISBN 978-82-471-3619-5
ISSN 1503-8181

Doctoral theses at NTNU, 2012:164

MILJØMERKET 241 TRYKKSAK 731

Printed by Skipnes Kommunikasjon as

Numerical Methods for Two-Phase Flow with Contact Lines

Claudio Walker

June 2012

Abstract

This thesis focuses on numerical methods for two-phase flows, and especially flows with a moving contact line. Moving contact lines occur where the interface between two fluids is in contact with a solid wall. At the location where both fluids and the wall meet, the common continuum descriptions for fluids are not longer valid, since the dynamics around such a contact line are governed by interactions at the molecular level. Therefore the standart numerical continuum models have to be adjusted to handle moving contact lines.

In the main part of the thesis a method to manipulate the position and the velocity of a contact line in a two-phase solver, is described. The Navier-Stokes equations are discretised using an explicit finite difference method on a staggered grid. The position of the interface is tracked with the level set method and the discontinuities at the interface are treated in a sharp manner with the ghost fluid method. The contact line is tracked explicitly and its dynamics can be described by an arbitrary function. The key part of the procedure is to enforce a coupling between the contact line and the Navier-Stokes equations as well as the level set method. Results for different contact line models are presented and it is demonstrated that they are in agreement with analytical solutions or results reported in the literature.

The presented Navier-Stokes solver is applied as a part in a multiscale method to simulate capillary driven flows. A relation between the contact angle and the contact line velocity is computed by a phase field model resolving the micro scale dynamics in the region around the contact line. The relation of the microscale model is then used to prescribe the dynamics of the contact line in the macro scale solver. This approach allows to exploit the scale separation between the contact line dynamics and the bulk flow. Therefore coarser meshes can be applied for the macro scale flow solver compared to global phase field simulations, reducing the overall computational coasts.

One of the major drawbacks of the level set method is that it does not conserve the mass of the fluids. The application of the conservative level set method (CLSM) offers a solution to this problem. Three of the attached articles address details concerning the implementation of the CLSM using a finite difference method. A finite difference discretisation of the CLSM based on stencils used in the Navier-Stokes solver is described and tested. Various methods to compute the curvature in the CLSM are assessed for the use in the ghost fluid method. It is shown that the reinitialisation of the CLSM can lead to spurious deformations of the interface, a stabilised constrained reinitialisation is developed in an attempt to prevent the interface from deforming.

Preface

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy (PhD) at the Norwegian University of Science and Technology (NTNU). The work presented in this thesis was done between June 2008 and June 2012 under the supervision of Professor Bernhard Müller under co-supervision of Professors Maria Fernandino and Tor Ytrehus.

The research was carried out at the Fluids Engineering group at the Department of Energy and Process Engineering, Faculty of Engineering Science and Technology, NTNU. During a stay of one month at Uppsala University in Sweden collaboration was conducted with Martin Kronbichler and Professor Gunilla Kreiss.

The research was funded by the Faculty of Engineering Science and Technology (IVT) at NTNU. The funding of the research position required that one forth of the time was spent for teaching duties at the Department of Energy and Process Engineering.

The main motivation for the research was to develop numerical methods for multiphase flows with moving contact lines.

Claudio Walker
Trondheim, June 2012

Acknowledgements

Now as all the fruits of four years of work are summarised in the thesis at hand, it is time to thank all the people who were involved in one way or an other. Without the support of you, it would not have been possible for my to get to this point.

First I want to thank my supervisor Professor Bernhard Müller, for giving me the opportunity to work on this project. During my studies I could always rely on his advise, support and motivation. I am grateful that he trusted me to develop my own ideas and I appreciated his open door whenever I had a question or felt that I am stuck on my way. I would also like to thank my co-supervisors Professors Tor Ytrehus and Maria Fernandino for accompanying my studies.

The exchange with colleagues is always important and inspiring. Especially, I would like to thank Joris Verschaeve for all the useful discussions on the different methods to keep track of the interface location as well as all the conversations about general topics related to both our PhD projects. I am also thankful to Karl Yngve Lervåg for the helpful exchanges on details concerning the implementation of the sharp interface solver. In addition I would like to thank Martin Larsson, Muhammad Asif Farooq, Halvor Lund and Jan Fredrik Helgaker, who regularly attended our group meetings, for their questions and ideas. A key part in the project was the collaboration with Martin Kronbichler and Professor Gunilla Kreiss at Department of Information Technology, Uppsala University. I highly appreciate their contribution to the project and all their support during my visit in Uppsala.

I enjoyed a good time while working at the institute and would like to express my gratitude to everybody who contributed to the pleasant working environment. Especially I want to mention all the fellow PhD students, the lecturers and teaching assistants involved in the lecture "Fluidmekanikk", and the administrative staff of the Department. Since I mentioned "Fluidmekanikk" I would like to voice my appreciation to all the students attending my exercise hours. Your non-verbal reactions each time I succeeded to explain something were a huge motivation and your critics helped to improve my teaching skills. The lunch and coffee brakes with Asif, Halvor, Ignat, Jan Fredrik, Joris, Karl Yngve, Kristian, Michael, Simen and Vagesh led to interesting, horizon broadening, amusing as well as harebrained conversations.

A huge thank you goes to my family and friends at home for their moral support during the last four years. I will forever be grateful to my parents for their support, motivation and trust during my education. The distance and my lack of time made it difficult to stay in touch at times, but my sister and Andrea Kuster provided always a warm welcome at home.

The final thanks go to all the friends I made in Trondheim. You played an important role making my life in Trondheim pleasant and enjoyable. In addition you provided me with the motivation and help to learn Norsk. I appreciated the open welcome at Trondhjems Seilforening and the good times we spent on the fjord with together. Terje Brasethvik and his family deserve a special thank for their generosity. Terje also kindly offered his help to proofread the manuscript.

Claudio Walker
Trondheim, July 2012

Thank you **Vielen Dank** **Tusen takk**

Contents of Thesis

Publications

[a] C. Walker and B. Müller. A conservative level set method for sharp interface multiphase flow simulation. In *Proceedings of ECCOMAS CFD 2010 Conference, Lisbon*, 2010.

[b] C. Walker. Curvature computation for a sharp interface method using the conservative level set method. In A. Eriksson and G. Tibert, editors, *Proceedings of NSCM-23*, pages 203–206, 2010.

[c] C. Walker and B. Müller. Contact line treatment with the sharp interface method. In B. Skallerud and H. Andersson, editors, *MekIT’11: Sixth national conference on Computational Mechanics*, pages 451–462. Tapir Academic Press Trondheim, 2011.

[d] C. Walker and B. Müller. Constrained reinitialisation of the conservative level set method. In J. Olsen and S. Johansen, editors, *Proceedings of the 8th International Conference on CFD in Oil & Gas, Metallurgical and Process Industries*, 2011.

[e] M. Kronbichler, C. Walker, G. Kreiss, and B. Müller. Microscale enhancement of macroscale modeling for capillary-driven contact line dynamics. Submitted, 2012.

[f] C. Walker and B. Müller. Contact line treatment with the sharp interface method. Submitted, 2012.

My contribution to thesis articles

I am the sole author of article [b]. As first author of articles [a], [c], [d] and [f], I carried out the work and wrote the papers with feedback form Bernhard Müller. The ideas for article [e] were developed in close discussion between the authors, I implemented the macro solver and performed the experiments with that model.

Chapter 1

Introduction

1.1 Background and Motivation

Multiphase flows are of importance in many natural and industrial processes. We all experience multiphase phenomena throughout our daily live. Examples range from opening the tap in the morning to raindrops making it hard to see through spectacles. In the process and energy industries multiphase flows play an important role. They can be crucial for the heat transport, for example in heat exchangers or boilers. In oil industry numerous examples of multiphase flows can be found from the well to the petrol station. Basically everywhere where liquids and gases are involved, some sort of multiphase phenomena are influencing the processes. The modelling and computational prediction of multiphase flows still offers many challenges.

The present thesis is concentrating on the numerical computation of multiphase flows which are in contact with solid walls. If an interface, which is the region separating two immiscible fluids, is in contact with a solid wall, a so called contact line is formed. Again there are numerous examples for flow situations containing contact lines. One of the most famous is the capillary rise of fluid in a small tube, where a denser fluid displaces a lighter one. As it is obvious from the capillary rise the physical effects which occur at a contact line can be important on the overall behaviour of the system. Despite that these kind of flows are very common in both industrial and natural processes, they are still not completely understood. Hence, they pose interesting challenges in terms of numerical simulations.

1.1.1 Methods for multiphase flow simulations

Here solely an overview over the methods which can be used to compute multiphase flows is given. The overview does not include methods where the averaged equations for multiphase systems are solved. Despite the fact that these methods enjoy a great popularity for dispersed flows we concentrate on methods, which are keeping track of the interface position and not just an averaged volume fraction of the different components. To develop methods for flows including contact lines, it is necessary to know the location where the interface between the fluids is in contact with a wall. A review over methods which can be used to track the position of the interface is given in [59].

Methods to model multiphase flows on fixed grids can be divided into two groups depending on the way they are modelling the interface. The first group assumes that the interface between the fluids is much smaller than any resolved length scale, such that the change in fluid properties at the interface can be treated as a discontinuity. Typically the interface is advected with the local fluid velocity and a number of different techniques exist to keep track of the interface position. The other group comprises methods where the interface has a finite extension. The fluid properties, such as density and viscosity, are changing rapidly but smoothly as the interface is crossed. Those are called phase field methods [3, 41] and they typically solve the coupled Cahn-Hilliard/Navier-Stokes equations. The Cahn-Hilliard equation is based on the free energy of an interface [9]. The phase field methods allow for contact line movement through diffusive interfacial fluxes across the interface, even if a no-slip boundary condition is applied [37], which is not the case for methods which do not resolve the interface. Since the behaviour of the interface is modelled using physical principles, it is necessary that the discretisation is resolving all the length scales of the interface. Those length scales are in general much smaller than the length scales of the flow, e.g. the size of a droplet. This resolution requirement is posing a severe restriction on the applicability of phase field methods for the simulation of problems including multiphase flow.

The advantage of methods not resolving the interface is, that they typically require less resolution, since there is no diffuse interface which needs to be resolved. Thus, those methods can be applied to problems on a larger scale. The methods, which do not resolve the interface solve the Navier-Stokes equations and apply appropriate interface conditions at the position of the interface. The main difference between those method is how they evolve the interface position in time [59, 67]. Tryggvason et al. [68] differentiate between front-tracking and front-capturing methods. Front-tracking methods use marker points that represent the interface and are tracked in a Lagrangian way. Since the particles tend to accumulate in certain areas of the flow, they have to be redistributed from time to time [67]. This redistribution can also be regarded as a kind of adaptivity. Therefore front-tracking methods tend to be able to resolve thin interface structures more accurately than front-capturing methods. Since there is a connectivity between the tracing particles topological changes have to be treated explicitly.

Front-capturing methods use an Eulerian scalar field to indirectly describe the interface position. The scalar field is evolved in time and it indicates the location of the interface. Prominent representatives for front-capturing methods are the volume of fluid method (VOF) [34] and the level set method (LSM) [65, 51, 60]. The VOF uses a volume fraction function which is either zero or one, depending on the volume fraction of a cell occupied by one of the fluids. Special advection schemes and interface reconstructions are applied such that the transition from zero to one is kept as accurate as possible. The VOF method conserves the mass of both fluids by construction. Instead of a volume fraction, the LSM uses a signed distance function. The signed distance function is continuous throughout the computational domain. The contour line where the distance function is zero is defined as the interface. One of the advantages of the LSM it enables a simple way to retrieve geometrical information of the interface such as the curvature and surface normal. However, the level set method has an important disadvantage because it does not conserve the mass of the two fluids [60]. The mass loss is most pronounced in regions with high curvature where the interface is not well resolved by the grid. Different approaches were developed to improve the mass conservation of the level set method. Examples include the conservative level set method (CLSM) [49] [50], the particle level set method (PLS) [24] or the coupled level set/volume-of-fluid (CLSVOF) [64]. The added complexity for both PLS and CLSVOF are significant. On the other hand the conservative level set method improves the mass conservation and keeps the simplicity of the original method. Front-capturing methods can handle topological changes in a natural way, since they use an Eulerian scalar function. If two interfaces approach each other and their distance becomes less than the grid spacing, front-capturing methods can no longer resolve both interfaces and they are merged. It has to be noted that the topological changes in front-capturing methods are not based on physical principles but rather a result of finite resolution of the computational grid.

The methods not resolving the interface, meet an additional challenge. Conventional discretisations can neither handle fluid properties which change instantly at the interface nor the singular surface tension force. Therefore the jump in the fluid properties and the surface tension is smeared artificially over multiple grid points [7, 12, 67]. An alternative is to develop numerical methods, which can handle sharp jumps in the solutions. Sharp interface methods applied for two-phase flow include the ghost-fluid method (GFM) [40], immersed interface method [45] and the extended finite element method [26]. The ghost-fluid method (GFM) [25] was introduced for compressible flows and later extended to incompressible two-phase flows [40]. It modifies the discretisation stencils to allow sharp jumps of the fluid properties at the interface as well as a sharp implementation of the surface tension force through a jump in the pressure. The majority of methods for modelling of multiphase flow including contact lines, where the interface is not resolved, smear the jumps and the surface tension over several grid points [8, 27, 63, 70, 55, 2, 17].

1.1.2 Contact line

One of the difficulties arising in multiphase flows is the moving contact line problem. This is the line where the interface between two different fluids intersects the surface of a solid substrate. In two dimensions the contact line is reduced to a point. There is not only a surface energy associated with the interface between the fluids, but there is also a surface energy associated with the interface between the solid and both of the fluids. In a system containing two fluids in contact with a solid wall, the fluids seek to assume a state where the energy is minimised. This leads to the well-known relation proposed by Young [71], which relates the three surface energies with the static contact angle.

If the fluids are not at rest the situation becomes more complicated. It was shown by Huh and Scriven [35] that the hydrodynamic equations in connection with the conventional no-slip boundary condition at the solid wall cannot be used to describe systems where the contact line is moving along a wall. It was demonstrated that the stresses are diverging at the line where the three phases meet [35]. Molecular dynamics (MD) simulations [42] show that there must be some sort of slip in the region around the contact line. Over the years several different models have been proposed to overcome the diverging stresses at the contact line. For an overview the reader is referred to review articles such as [16, 44, 6]. Most of the analytical descriptions using a hydrodynamic descriptions rely on the following three assumptions [6]. First, the Capillary number $\mathrm{Ca} = \frac{\mu U}{\sigma}$, where μ is the dynamic viscosity, U a characteristic velocity and σ the surface tension, is assumed to be small. Second, the inertial effects are considered to be negligible. And third, the surfaces are assumed to be heterogeneous such that there is no contact angle hysteresis. The lubrication theory, where the hydrodynamic equations are simplified and expressed with the help of the distance between the interface and the wall h, is used to describe the flow in the vicinity of the contact line. Then it is assumed that there is some kind of slip in a region with an extent of L_s around the contact line. The length scale L_s is associated with interactions at the molecular scale and is of the order of nano meters [5]. The result of those analyses is some variation of the Cox-Voinov law [6]

$$h'(x) = \theta_m + 9\mathrm{Ca}\ln(x/L_s), \tag{1.1}$$

where x is the distance from the contact point and θ_m is the microscopic contact angle. It is often assumed that θ_m is equal to the static contact angle θ_s [6]. Because the interface slope and therefore its angle with the wall vary logarithmicly as the contact line is approached, it is difficult to measure θ_m. This slope (1.1) of the interface shape is valid close to the contact point and can be matched to analytical or numerical solutions of the flow away from the interface. But the distance where this matching is done is usually still much smaller than the grid size of typical multiphase flow simulations.

While phase field methods can be used to simulate flows with moving contact lines [37, 69, 19] without the introduction of a slip boundary condition, it was demonstrated by Yue et al. [72] that the interface thickness must be of the order of the physical thickness of the interface which is of the same order as L_s. That means around the contact line a grid spacing in the order of L_s is required in order to obtain the sharp interface limit. Qian et al. [53] proposed a slip boundary condition for phase field methods.

As it is often not possible to resolve the length scales at which slip happens, most multiphase flow simulations relieve the no-slip boundary condition around contact lines and force the interface to intersect the wall at an angle $\theta_{\text{num}} = F(\text{Ca},\dots)$. How F should be chosen is still a matter of debate [68]. Examples using various forms of F are discussed in [8, 4, 57, 27, 46, 10]. However, there are a few methods which do not force the contact angle. Deganello et al. [17] manipulate the surface tension force close to the wall to control the motion of the contact line. Spelt [63] tracks a contact point which is moving with a velocity $u_{\text{CP}} = f(\theta_{\text{CP}})$, where θ_{CP} is the angle between the interface and the wall. The contact point treatment proposed in section 3.4 is based on [63] but it is adapted for a multiphase solver using the GFM.

1.1.3 Multiscale methods for contact lines

An alternative approach to overcome the singularity at the contact line is the application of multiscale models. The idea is to model the bulk of the domain with conventional continuum models and couple it to a more detailed molecular model around the contact line. The first application of a multiscale method to two-phase flows was developed by [28]. To couple the MD to the continuum description the domains of the two models overlap each other. In every time step multiple iterations are computed in each domain using the results of the other model as boundary conditions in the overlapping region until the two models converge in the shared region. A more efficient way to couple the two models is the so-called heterogeneous multiscale method (HMM) [22, 23]. HMM assumes a scale separation between the bulk flow and the flow around the contact points. Therefore the macroscopic model does not need to resolve the dynamics around the contact point. The macro model should offer a procedure to apply integral values describing the contact line dynamics. Those integral values are measured in the micro scale simulation around the contact point, which also requires some information from the macro model. This method was successfully applied to two-phase flows with contact lines [54]. Both implementations [28, 54] of multiscale methods for two-phase flows were applied to two immiscible fluids in Couette or Poisseuile flows. In both cases the density and viscosity were the same for the two fluids.

1.2 Goal and contribution of the present thesis

The goal of the PhD. project was to develop a multiscale method to simulate two-phase flows with a moving contact line. A multiphase solver, which is resolving the macroscopic processes should be used to model the bulk of the domain, while an appropriate micro scale model should resolve the relevant length scales around the contact line. The spacial scale separation should be exploited to couple the two models at both scales using the HMM approach. To make a step towards the application of such a multiscale method for real flow problems it was desired that both the macro and micro scale models should be able to handle fluids with different viscosities and densities. Further it was desired that a sharp interface method should be employed in the macro model, i.e. the surface tension and the jump of the fluid parameters should be treated in a sharp manner. Most of the effort was done to develop a suitable two-phase solver, where the motion of the contact point can be controlled using the input of the micro model. Therefore the details of the macro solver form the main part of the thesis. For details on the other contributions the reader is referred to the attached publications.

The main contributions of the present thesis are:

- Assessment of an alternative curvature computation of the conservative level set method. The details are given in publication [b].

- Stabilisation of the high order constrained reinitialisation for level set functions with large gradients. The details are given in publication [d].

- Implementation and verification of a two-phase solver using a sharp interface method. The details are given in section 3.1 and the publications [c] and [f].

- Development of a contact point treatment for the sharp interface method, which allows to track a contact point. The motion of the contact point can be described by an arbitrary function. The contact point treatment includes a method to compute convergent curvatures (see section 3.4).

- Development of a multiscale method for capillary driven contact line dynamics. The details are given in publication [e].

1.3 Outline of the thesis

The remainder of the thesis is organised as follows. In chapter 2 an overview over the derivation of the governing equations for two-phase flow is given. The chapter starts with the single phase equations which then are used to derive the interface conditions between two immiscible incompressible fluids with constant densities and viscosities. The concept of surface tension is introduced and added to the interface conditions. The chapter closes with the reformulation of the interface conditions such that they can be used by the ghost fluid method.

Chapter 3 gives a detailed description of the numerical methods used to discretise the Navier-Stokes equations for multiphase flow. The spacial discretisation for the single phase equations are introduced first, before the level set method, which is used to track the interface, and its implementation are described. After the introduction of the Ghost fluid method the procedure for the contact point tracking and its adaption for the ghost fluid method are explained in detail in section 3.4.

Results from numerical experiments verifying the methods presented in the theses are shown in chapter 4. Damped surface waves are computed to verify the implementation of the two-phase solver (section 4.1). Alternative approximations for the advective terms and the jump conditions were evaluated using the same example. The accuracy of the contact point treatment was tested using a capillary rise (section 4.2.1). The chapter concludes with three examples involving dynamic contact point behaviour. For two of the examples, i.e. capillary driven channel flow (section 4.2.3) and advancing droplet (section 4.2.4), the contact point dynamics is prescribed using the multiscale method from [e].

The thesis is closed with chapter 5, where the conclusions are drawn and some suggestions for further work are given.

The research articles are summarised in chapter 6 and attached at the end of the thesis.

Chapter 2

Governing equations

In this chapter a derivation of the mass and momentum conservation laws for multiphase flow is given. In multiphase flow the single phase equations must hold in each individual fluid, as well as for the entire fluid domain containing the two different fluids. The conditions at the interface between two fluids can be derived using the integral conservation equations. The discretisation which is presented in the next chapter is using the differential form of the conservation laws. Therefore, we give first an overview of the derivation of the integral mass and momentum conservation laws and use those to get the differential form as well as the interface conditions. The derivations presented here are based on notes form the lecture "Modelling of Multiphase flows" which was held by Tor Ytrehus at NTNU [39], which in turn is partly based on the book by Thompson [66].

2.1 Conservation laws for single phase flow

2.1.1 Mass conservation

The temporal change of a quantity χ inside a general moving control volume $\mathcal{V}^*$ is given by the generalised Leibniz rule of integration (see e.g. [43])

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{V}^*} \chi \,\mathrm{d}V = \int_{\mathcal{V}^*} \frac{\partial \chi}{\partial t}\,\mathrm{d}V + \int_{\mathcal{A}^*} \chi \mathbf{n}\cdot\mathbf{b}\,\mathrm{d}A. \tag{2.1}$$

Where $\mathcal{A}^*$ is the surface of the control volume with an outward pointing unit normal $\mathbf{n}$ and local velocity $\mathbf{b}$. The mass of a given material fluid volume $\mathcal{V}$ is given as $\int_{\mathcal{V}} \rho \,\mathrm{d}V$, where ρ is the local fluid density. Since mass is neither created nor destroyed we must have $\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{V}} \rho \,\mathrm{d}V = 0$. If we set $\mathcal{V}^* = \mathcal{V}$, $\mathcal{A}^* = \mathcal{A}$ and $\mathbf{b} = \mathbf{u}$ (i.e. the velocity of the control volume is equal to the local fluid velocity) in equation (2.1) we get:

$$0 = \int_{\mathcal{V}} \frac{\partial \rho}{\partial t} \, \mathrm{d}V + \int_{\mathcal{A}} \rho \mathbf{u} \cdot \mathbf{n} \, \mathrm{d}A. \tag{2.2}$$

Which can be rewritten with the help of the divergence theorem

$$0 = \int_{\mathcal{V}} \frac{\partial \rho}{\partial t} \, \mathrm{d}V + \int_{\mathcal{V}} \nabla \cdot (\rho \mathbf{u}) \, \mathrm{d}V = \int_{\mathcal{V}} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \, \mathrm{d}V. \tag{2.3}$$

This equation must hold for an arbitrary material volume $\mathcal{V}$, which is only true if the integrand is zero. We get therefore the differential form of the mass conservation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0. \tag{2.4}$$

In the case of an incompressible fluid the density is constant and the differential form of the mass conservation reduces to

$$\nabla \cdot \mathbf{u} = 0. \tag{2.5}$$

Which means that the velocity field of an incompressible fluid must be divergence free.

In order to derive the conditions at a fluid interface we need the mass conservation equations for a general moving control volume $\mathcal{V}^*$. For this purpose we replace χ in equation (2.1) by the density ρ

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathcal{V}^*} \rho \, \mathrm{d}V = \int_{\mathcal{V}^*} \frac{\partial \rho}{\partial t} \, \mathrm{d}V + \int_{\mathcal{A}^*} \rho \mathbf{n} \cdot \mathbf{b} \, \mathrm{d}A. \tag{2.6}$$

Let us now assume that at a given time t_0 the material fluid volume $\mathcal{V}$ coincides with the general moving control volume $\mathcal{V}^*$. At this time t_0 we subtract equation (2.2) from (2.6) and we get the mass conservation for a general moving control volume:

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathcal{V}^*} \rho \, \mathrm{d}V + \int_{\mathcal{A}^*} \rho \left(\mathbf{u} - \mathbf{b} \right) \cdot \mathbf{n} \, \mathrm{d}A = 0. \tag{2.7}$$

2.1.2 Momentum conservation

The forces acting in a fluid can be divided into three classes [43]. Body forces are acting on the volume of the fluid. Here the body forces per unit mass will be denoted by $\mathbf{g}$ and the body force acting on the fluid volume becomes $\int_{\mathcal{V}} \rho \mathbf{g} \,\mathrm{d}V$. The second class comprises surface forces, which act on the surface of a fluid volume. They can be divided into forces acting normal on the surface (normal forces, e.g. pressure) and tangential to the surface (shear forces). The surface forces acting on a fluid volume are given as $\int_{\mathcal{A}} \mathbf{T} \cdot \mathbf{n} \,\mathrm{d}A$, where $\mathbf{T}$ is the stress tensor. The last class comprises line forces, which act along lines (in 2 dimensions) or surfaces (in three dimensions). An example is the surface tension which acts only at the interface between two fluids. They are singular in the sense, that they are confined to specific lines or surfaces. In some notations (e.g. [68, chapter 2.5]) they enter the equations of motion using a delta function i.e. $\int_{\mathcal{V}} \hat{\mathbf{f}} \delta \left(\mathbf{x} - \mathbf{x}_{\mathbf{\Gamma}}\right) \,\mathrm{d}V$, where $\hat{\mathbf{f}}$ denotes the line force, $\mathbf{x}_{\mathbf{\Gamma}}$ is the location of the line where the force is active and δ denotes the delta function. In other notations the line forces do not appear directly in the conservation equations, but are considered through appropriate boundary conditions [43, chapter 4.5]. Here we are using the second approach, and we will introduce the surface tension in the interface conditions.

The momentum of a fluid volume $\mathcal{V}$ is given as $\int_{\mathcal{V}} \rho \mathbf{u} \,\mathrm{d}V$. According to Newton's second law of motion the temporal change of the momentum for a given fluid volume is equal to the forces acting on it

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathcal{V}} \rho \mathbf{u} \,\mathrm{d}V = \int_{\mathcal{A}} \mathbf{T} \cdot \mathbf{n} \,\mathrm{d}A + \int_{\mathcal{V}} \rho \mathbf{g} \,\mathrm{d}V. \tag{2.8}$$

To get a differential form of the momentum conservation equation we take the Leibniz rule (2.1) for a material volume $\mathcal{V}$, such that $\mathbf{b} = \mathbf{u}$ and replace χ by $\rho\varphi$. Then the divergence theorem is applied to the surface integral and the product rule is used to get

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{V}}\rho\varphi\,\mathrm{d}V &= \int_{\mathcal{V}}\frac{\partial}{\partial t}(\rho\varphi)\,\mathrm{d}V + \int_{\mathcal{A}}\rho\varphi\mathbf{n}\cdot\mathbf{u}\,\mathrm{d}A\\ &= \int_{\mathcal{V}}\frac{\partial}{\partial t}(\rho\varphi)\,\mathrm{d}V + \int_{\mathcal{V}}\nabla\cdot(\rho\varphi\mathbf{u})\,\mathrm{d}A\\ &= \int_{\mathcal{V}}\rho\left(\frac{\partial\varphi}{\partial t}+(\mathbf{u}\cdot\nabla)\,\varphi\right)\,\mathrm{d}V + \int_{\mathcal{V}}\varphi\left(\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\mathbf{u})\right)\,\mathrm{d}V\\ &= \int_{\mathcal{V}}\rho\left(\frac{\partial\varphi}{\partial t}+(\mathbf{u}\cdot\nabla)\,\varphi\right)\,\mathrm{d}V = \int_{\mathcal{V}}\rho\frac{\mathrm{D}\varphi}{\mathrm{D}t}\,\mathrm{d}V. \qquad (2.9)\end{aligned}$$

The the mass conservation (2.4) and the definition of the substantial derivative $\frac{\mathrm{D}}{\mathrm{D}t}=\frac{\partial}{\partial t}+(\nabla\cdot\mathbf{u})$ were used in the last two steps. Inserting this result into equation (2.8) and applying the divergence theorem yields for $\varphi=\mathbf{u}$

$$\int_{\mathcal{V}}\rho\left(\frac{\partial\mathbf{u}}{\partial t}+(\mathbf{u}\cdot\nabla)\,\mathbf{u}\right)\,\mathrm{d}V = \int_{\mathcal{V}}\nabla\cdot\mathbf{T}\,\mathrm{d}V + \int_{\mathcal{V}}\rho\mathbf{g}\,\mathrm{d}V. \qquad (2.10)$$

Again the equation must be fulfilled for any fluid volume $\mathcal{V}$ and therefore we arrive at the differential form of the momentum equation

$$\rho\left(\frac{\partial\mathbf{u}}{\partial t}+(\mathbf{u}\cdot\nabla)\,\mathbf{u}\right) = \nabla\cdot\mathbf{T}+\rho\mathbf{g}. \qquad (2.11)$$

By replacing χ with $\rho\mathbf{u}$ in equation (2.1) and subtracting it from equation (2.8) the integral form of the momentum conservation which holds for a general moving control volume $\mathcal{V}^*$ is obtained

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{V}^*}\rho\mathbf{u}\,\mathrm{d}V+\int_{\mathcal{A}^*}\rho\mathbf{u}\,(\mathbf{u}-\mathbf{b})\cdot\mathbf{n}\,\mathrm{d}A = \int_{\mathcal{A}^*}\mathbf{T}\cdot\mathbf{n}\,\mathrm{d}A+\int_{\mathcal{V}^*}\rho\mathbf{g}\,\mathrm{d}V. \quad (2.12)$$

2.1.3 Constitutive equation for a Newtonian fluid

The stresses in a fluid at rest only act in normal direction of a surface and are independent of the angular orientation of this surface. The stress tensor must take the following form $\mathbf{T}=-p\mathbf{I}$, where p is the fluid pressure and $\mathbf{I}$ the unit tensor. The negative sign is because the pressure acts in opposite direction of the surface unit normal $\mathbf{n}$.

In a moving fluid, the stress tensor contains additional components, which in a Newtonian fluid are proportional to the strain rate. It can be shown (for details see [43, chapter 4.10]) that the stress tensor for a Newtonian fluid must take the following form

$$\mathbf{T} = -p\mathbf{I} + \mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) + \lambda \left(\nabla \cdot \mathbf{u}\right) \mathbf{I}, \tag{2.13}$$

where μ is the dynamic viscosity and λ the second viscosity coefficient. If we consider an incompressible fluid the velocity field is divergence free and the stress tensor is reduced to

$$\mathbf{T} = -p\mathbf{I} + \boldsymbol{\tau}, \tag{2.14}$$

where the viscous shear stress tensor is given as

$$\boldsymbol{\tau} = \mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right). \tag{2.15}$$

Substituting equation (2.14) into equation (2.11) and using the mass conservation (2.4) we obtain the single phase Navier-Stokes equations for incompressible fluids

$$\nabla \cdot \mathbf{u} = 0 \tag{2.16}$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \left(\mathbf{u} \cdot \nabla\right) \mathbf{u}\right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{g}, \tag{2.17}$$

where μ is assumed to be constant.

2.2 Interface conditions

Immiscible multiphase flow can be regarded as multiple domains with single phase flow which are in contact with each other. The equations of motion derived in the previous section must hold in each phase. In addition they must also be fulfilled in the entire domain containing the multiphase flow. The single phase equations can be used to derive conditions at the interface between two fluids.

2.2.1 Conservation of mass

We consider a control volume $\mathcal{V}^*$ which contains two fluids separated by an interface as illustrated in Figure 2.1. We assume that $\mathcal{V}^*$ moves with the interface such that its thickness normal to the interface is always 2ϵ. The surface of the control volume is $\mathcal{A}^*$ and the velocity of the control volume surface is denoted as $\mathbf{b}$. The interface divides the control volume $\mathcal{V}^*$ into two sub volumes $\mathcal{V}_1^*$ and $\mathcal{V}_2^*$. We use Σ_1 and Σ_2 to denote the surfaces of $\mathcal{V}_1^*$ and $\mathcal{V}_2^*$, respectively, excluding the surface Σ_S, which is the interface inside the control volume. Such that we have $\mathcal{A}^* = \Sigma_1 \cup \Sigma_2$.

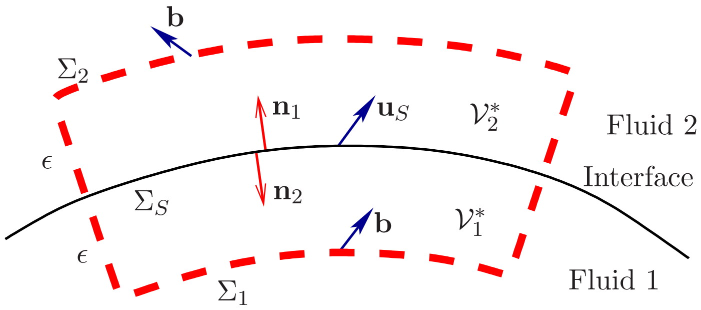

Figure 2.1: Control volume containing an fluid fluid interface.

The mass conservation laws (2.7) for the three control volumes read:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{V}^*} \rho \,\mathrm{d}V + \int_{\mathcal{A}^*} \rho \left(\mathbf{u} - \mathbf{b}\right) \cdot \mathbf{n} \,\mathrm{d}A = 0 \tag{2.18}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{V}_1^*} \rho_1 \,\mathrm{d}V + \int_{\Sigma_1 \cup \Sigma_S} \rho_1 \left(\mathbf{u} - \mathbf{b}\right) \cdot \mathbf{n} \,\mathrm{d}A = 0 \tag{2.19}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{V}_2^*} \rho_2 \,\mathrm{d}V + \int_{\Sigma_2 \cup \Sigma_S} \rho_2 \left(\mathbf{u} - \mathbf{b}\right) \cdot \mathbf{n} \,\mathrm{d}A = 0 \tag{2.20}$$

The velocity of the control volume surface along the interface Σ_S is equal to the interface velocity $\mathbf{u}_S$. In addition along Σ_S we have $\mathbf{n}_2 = -\mathbf{n}_1$. Equations (2.19) and (2.20) are subtracted from equation (2.18) to yield:

$$\int_{\Sigma_S} \left(\rho_1 \left(\mathbf{u}_1 - \mathbf{u}_S\right) \cdot \mathbf{n}_1 - \rho_2 \left(\mathbf{u}_2 - \mathbf{u}_S\right) \cdot \mathbf{n}_1\right) \,\mathrm{d}A = 0. \tag{2.21}$$

This must hold for any control volume $\mathcal{V}^*$ containing the interface and therefore we can write:

$$\rho_1 \left(\mathbf{u}_1 - \mathbf{u}_S\right) \cdot \mathbf{n}_1 - \rho_2 \left(\mathbf{u}_2 - \mathbf{u}_S\right) \cdot \mathbf{n}_1 = 0. \tag{2.22}$$

The quantity $\dot{m}_1 = \rho_1 \left(\mathbf{u}_1 - \mathbf{u}_S\right) \cdot \mathbf{n}_1$ is the mass flow leaving fluid 1.

Assuming there is no mass exchange between the two fluids (e.g. there is no evaporation and the fluids are immiscible) we have $\dot{m}_1 = 0$ and therefore $\mathbf{u}_S \cdot \mathbf{n}_1 = \mathbf{u}_1 \cdot \mathbf{n}$. Thus the interface condition is further reduced to

$$[\![\mathbf{u}]\!] \cdot \mathbf{n} = 0 \tag{2.23}$$

where $[\![\ldots]\!]$ denotes the jump at the interface with position $\mathbf{x}_\Gamma$, i.e. $[\![\mathbf{u}]\!] = \mathbf{u}_2(\mathbf{x}_\Gamma) - \mathbf{u}_1(\mathbf{x}_\Gamma)$. This means that the interface normal component of the velocity is continuous across the interface, and the normal interface velocity is equal to the normal component of the velocities at the interface.

2.2.2 Kinematic condition at the interface

In addition to the condition that the mass must be conserved at the interface we also have a kinematic condition for viscous fluids, which states that the tangential velocity component of two neighbouring fluids must be equal, i.e. $[\![\mathbf{u}]\!] \cdot \mathbf{t} = 0$ where $\mathbf{t}$ is a vector tangential to the interface with unit length. Together with equation (2.23) the jump conditions for the velocity at the interface read:

$$[\![\mathbf{u}]\!] = 0 \tag{2.24}$$

$$[\![\nabla \mathbf{u}]\!] \cdot \mathbf{t} = 0. \tag{2.25}$$

This implies that the velocity $\mathbf{u}$ and its tangential derivative are continuous over the interface. The latter follows automatically from (2.24). If both phases are incompressible we get the trivial identity $[\![\nabla \cdot \mathbf{u}]\!] = 0$. Using the identity $\nabla \cdot \mathbf{u} = \mathbf{n} \cdot \nabla \mathbf{u} \cdot \mathbf{n} + \mathbf{t} \cdot \nabla \mathbf{u} \cdot \mathbf{t}$ and equation (2.25) we get:

$$[\![\mathbf{n} \cdot \nabla \mathbf{u} \cdot \mathbf{n}]\!] = 0, \tag{2.26}$$

which means that the normal component of the normal derivative of the velocity field is continuous across the interface.

2.2.3 Conservation of momentum

To derive the interface conditions for the stresses we use again the same control volumes as described in section 2.2.1 and illustrated in Figure 2.1. The momentum conservation laws (2.12) for the control volumes read:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{V}^*} \rho\mathbf{u}\,\mathrm{d}V + \int_{\mathcal{A}^*} \rho\mathbf{u}\,(\mathbf{u}-\mathbf{b})\cdot\mathbf{n}\,\mathrm{d}A = \int_{\mathcal{A}^*} \mathbf{T}\cdot\mathbf{n}\,\mathrm{d}A + \int_{\mathcal{V}^*} \rho\mathbf{g}\,\mathrm{d}V \tag{2.27}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{V}_1^*} \rho_1\mathbf{u}_1\,\mathrm{d}V + \int_{\Sigma_1\cup\Sigma_S} \rho_1\mathbf{u}\,(\mathbf{u}-\mathbf{b})\cdot\mathbf{n}\,\mathrm{d}A = \int_{\Sigma_1\cup\Sigma_S} \mathbf{T}\cdot\mathbf{n}\,\mathrm{d}A + \int_{\mathcal{V}_1^*} \rho_1\mathbf{g}\,\mathrm{d}V \tag{2.28}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{V}_2^*} \rho_2\mathbf{u}_2\,\mathrm{d}V + \int_{\Sigma_2\cup\Sigma_S} \rho_2\mathbf{u}\,(\mathbf{u}-\mathbf{b})\cdot\mathbf{n}\,\mathrm{d}A = \int_{\Sigma_2\cup\Sigma_S} \mathbf{T}\cdot\mathbf{n}\,\mathrm{d}A + \int_{\mathcal{V}_2^*} \rho_2\mathbf{g}\,\mathrm{d}V. \tag{2.29}$$

Repeating the procedure from section 2.2.1, assuming no mass exchange across the interface and using the constitutive equation for an incompressible Newtonian fluid (equation (2.14)) the interface condition for the stresses is obtained

$$[\![-p\mathbf{I}+\boldsymbol{\tau}]\!]\cdot\mathbf{n} = 0. \tag{2.30}$$

This means that the stresses are continuous across the interface.

2.2.4 Surface tension

The interface between two fluids behaves like a flexible membrane. To increase the area of the interface work has to be done, i.e. there is a surface energy which is proportional to the area of the interface. The surface tension σ represents the stretching work that needs to be done to increase the interface area by a unit amount [11]. Therefore we can often observe that small droplets assume a spherical shape in order to reduce the surface area and thus the surface energy. The source of the surface tension lies at the atomic scale. At the atomic scale all fluid molecules interact with each other through attractive and repulsive forces. For a single fluid molecule which is immersed in a fluid at rest, the averaged resulting force is zero. Since the interaction forces between two fluid molecules of different type are not the same as the interactions between identical fluids, we have a nonzero force acting on a fluid molecule close

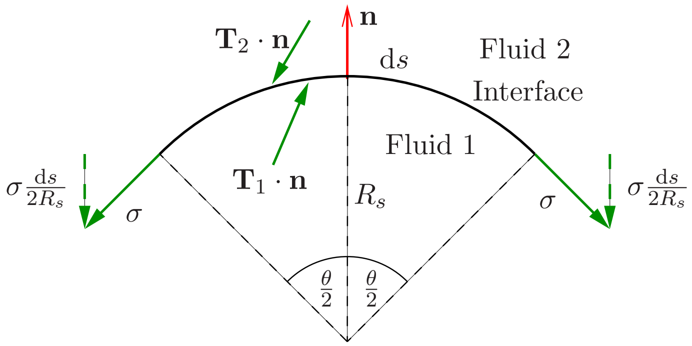
$\mathbf{n}$
$\mathbf{T}_2 \cdot \mathbf{n}$
$\mathrm{d}s$
Fluid 2
Interface
Fluid 1
$\mathbf{T}_1 \cdot \mathbf{n}$
R_s
$\sigma \frac{\mathrm{d}s}{2R_s}$
σ
σ
$\sigma \frac{\mathrm{d}s}{2R_s}$
$\frac{\theta}{2}$
$\frac{\theta}{2}$

Figure 2.2: Forces on a small interface area including surface tension.

to an interface between two fluids. More information on the origin of surface tension can be found in textbooks on the subject e.g. [1, 36].

Figure 2.2 shows a cross section trough a small interface element, $\mathrm{d}s$ is the length of the interface element in the paper plane and $\mathrm{d}l$ is the length of the interface element perpendicular to the paper plane. The radius of the surface element in the paper plane is R_s and the normal vector $\mathbf{n}$ is pointing from fluid 1 to fluid 2. The force from the surface tension will pull at both ends in tangential direction of the surface element. Its magnitude per length into the paper plane is σ at both end points if the surface tension is constant along the interface, i.e. there are no temperature gradients and no surfactants present. The net force of the surface tension in tangential direction vanishes and in normal direction the resulting force from the surface tension is $-\sigma \frac{\mathrm{d}s}{R_s}\mathbf{n}$. Therefore the balance of forces on the interface requires

$$\mathbf{T}_1 \cdot \mathbf{n} - \mathbf{T}_2 \cdot \mathbf{n} - \frac{\sigma}{R_s}\mathbf{n} = 0. \tag{2.31}$$

The same argument can be repeated in a plane which is perpendicular to the paper and parallel to the normal vector in Figure 2.2. Introducing the local curvature of the fluid element $\kappa = \frac{1}{R_s} + \frac{1}{R_l}$, where R_s and R_l are the radii of the interface measured in two perpendicular planes, and using the constitutive equation for an incompressible Newtonian fluid the interface conditions (2.30) becomes

$$[\![-p\mathbf{I} + \boldsymbol{\tau}]\!] \cdot \mathbf{n} = \kappa\sigma\mathbf{n}. \tag{2.32}$$

The inclusion of the surface tension in the interface condition leads to a discontinuity in the normal stresses across the interface, while the tangential stresses remain continuous.

2.2.5 Reformulation of the interface conditions

In order to use the interface conditions for the discretisation they have to be rewritten. In the ghost fluid method the jump in the pressure $[\![p]\!]$ and the jump in the derivatives of the velocity components along the coordinate lines $[\![\mu\nabla\mathbf{u}]\!]$ are required. Two versions of the reformulation procedure, which differ in the notation, can be found in [29] and [40].

First the jump condition for the pressure is obtained by taking the inner product of the jump condition including surface tension (2.32) with the interface normal $\mathbf{n}$ and using the definition of $\boldsymbol{\tau}$ (2.15)

$$[\![p]\!] = [\![\mathbf{n} \cdot 2\mu\nabla\mathbf{u} \cdot \mathbf{n}]\!] - \sigma\kappa = 2\,[\![\mu]\!]\,\mathbf{n} \cdot \nabla\mathbf{u} \cdot \mathbf{n} - \sigma\kappa, \tag{2.33}$$

where equation (2.26) was used for the last step.

Equation (2.25) is used to write

$$[\![\mu\nabla\mathbf{u}]\!] \cdot \mathbf{t} = [\![\mu]\!]\,\nabla\mathbf{u} \cdot \mathbf{t}. \tag{2.34}$$

The inner product of the jump condition (2.32) with the interface tangent $\mathbf{t}$ becomes:

$$\begin{aligned} 0 &= \left[\!\!\left[\mathbf{t} \cdot \mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) \cdot \mathbf{n}\right]\!\!\right] \\ &= \mathbf{t} \cdot [\![\mu\nabla\mathbf{u}]\!] \cdot \mathbf{n} + \mathbf{n} \cdot [\![\mu]\!]\,\nabla\mathbf{u} \cdot \mathbf{t}, \end{aligned} \tag{2.35}$$

where (2.25) was used in the last equality. Decomposing $[\![\mu\nabla\mathbf{u}]\!] \cdot \mathbf{n}$ into its normal and tangential components and then using the equation above yields

$$\begin{aligned} [\![\mu\nabla\mathbf{u}]\!] \cdot \mathbf{n} &= (\mathbf{n} \cdot [\![\mu\nabla\mathbf{u}]\!] \cdot \mathbf{n})\,\mathbf{n} + (\mathbf{t} \cdot [\![\mu\nabla\mathbf{u}]\!] \cdot \mathbf{n})\,\mathbf{t} \\ &= (\mathbf{n} \cdot [\![\mu\nabla\mathbf{u}]\!] \cdot \mathbf{n})\,\mathbf{n} - (\mathbf{n} \cdot [\![\mu]\!]\,\nabla\mathbf{u} \cdot \mathbf{t})\,\mathbf{t} \\ &= (\mathbf{n} \cdot [\![\mu]\!]\,\nabla\mathbf{u} \cdot \mathbf{n})\,\mathbf{n} - (\mathbf{n} \cdot [\![\mu]\!]\,\nabla\mathbf{u} \cdot \mathbf{t})\,\mathbf{t}, \end{aligned} \tag{2.36}$$

where equation 2.26 was used in the last step. The jumps in the viscous terms in an interface normal coordinate system can be obtained by taking the inner product of equations (2.34) and (2.36) with the normal and tangential vector respectively

$$\begin{aligned}\mathbf{n}\cdot[\![\mu\nabla\mathbf{u}]\!]\cdot\mathbf{n} &= [\![\mu]\!]\,\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n}\\ \mathbf{n}\cdot[\![\mu\nabla\mathbf{u}]\!]\cdot\mathbf{t} &= [\![\mu]\!]\,\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{t}\\ \mathbf{t}\cdot[\![\mu\nabla\mathbf{u}]\!]\cdot\mathbf{n} &= -\,[\![\mu]\!]\,\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{t}\\ \mathbf{t}\cdot[\![\mu\nabla\mathbf{u}]\!]\cdot\mathbf{t} &= [\![\mu]\!]\,\mathbf{t}\cdot\nabla\mathbf{u}\cdot\mathbf{t},\end{aligned}\tag{2.37}$$

which can be rewritten in coordinate free form:

$$\begin{aligned}[\![\mu\nabla\mathbf{u}]\!] = {}& [\![\mu]\!]\,\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n}(\mathbf{n}\otimes\mathbf{n}) + [\![\mu]\!]\,\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{t}(\mathbf{n}\otimes\mathbf{t})\\ &- [\![\mu]\!]\,\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{t}(\mathbf{t}\otimes\mathbf{n}) + [\![\mu]\!]\,\mathbf{t}\cdot\nabla\mathbf{u}\cdot\mathbf{t}(\mathbf{t}\otimes\mathbf{t}),\end{aligned}\tag{2.38}$$

where $\otimes$ denotes the dyadic product. The property $\mathbf{a}\cdot(\mathbf{b}\otimes\mathbf{c}) = (\mathbf{a}\cdot\mathbf{b})\mathbf{c}$ of the dyadic product, can be applied to the gradient, i.e. $\mathbf{a}\cdot(\nabla\mathbf{b}) = \nabla(\mathbf{a}\cdot\mathbf{b})$. Hence, the normal or tangent can be moved inside the differential operator. This is how equation (2.38) can be rewritten in the form used by [29, equation (2.28)]

$$\begin{aligned}[\![\mu\nabla\mathbf{u}]\!] = {}& [\![\mu]\!]\,\mathbf{n}\cdot\nabla(\mathbf{u}\cdot\mathbf{n})(\mathbf{n}\otimes\mathbf{n}) + [\![\mu]\!]\,\mathbf{t}\cdot\nabla(\mathbf{u}\cdot\mathbf{n})(\mathbf{n}\otimes\mathbf{t})\\ &- [\![\mu]\!]\,\mathbf{t}\cdot\nabla(\mathbf{u}\cdot\mathbf{n})(\mathbf{t}\otimes\mathbf{n}) + [\![\mu]\!]\,\mathbf{t}\cdot\nabla(\mathbf{u}\cdot\mathbf{t})(\mathbf{t}\otimes\mathbf{t}).\end{aligned}\tag{2.39}$$

Chapter 3

Numerical methods

In this chapter the numerical methods will be described to solve the equations governing multiphase flow presented in the previous chapter. The method to solve the Navier-Stokes equations for two-phases is based on the method presented by Kang et al. [40]. First the procedure to solve the single phase Navier-Stokes equations (NS) is presented. Then the level set method, which is used to track the interface is introduced. And finally the ghost fluid method (GFM) is described. The GMF implements the jump conditions form section 2.2, and thus extends the single phase solver to a two-phase solver. At the end of the chapter the methods to handle a contact point are described in detail.

3.1 Navier-Stokes solver

3.1.1 Projection method

For incompressible flow the mass conservation reduces to a constraint on the velocity field, i.e. the velocity field must be divergence free, cf. equation (2.5). This constraint is changing the nature of the equation system. While the momentum conservation equation forms a parabolic system, the incompressible Navier-Stokes equations are not purely parabolic anymore. They contain an elliptic part, because the divergence of (2.17) together with (2.16) yields a Poisson equation for the pressure. This dictates that appropriate numerical methods have to be used to solve the coupled equations. One of the possible strategies to solve the incompressible Navier-Stokes equations are projection methods. Other alternatives can be found for example in [20]. The idea of the projection methods is to advance the velocity field in time without taking care of the divergence free constraint and then adjust the resulting intermediate velocity field such that it has zero divergence. Thus, the velocity field is projected on the space of solenoidal vector fields.

The projection methods are based on the Helmholtz-Hodge theorem, which states that an arbitrary vector field **a** can be decomposed into a divergence free or solenoidal part and a rotation free part. Such that it can be written

$$\mathbf{a} = \mathbf{a}^d + \nabla\varphi, \tag{3.1}$$

where $\nabla \cdot \mathbf{a}^d = 0$. Taking the divergence of equation (3.1) yields

$$\nabla \cdot \mathbf{a} = \nabla^2\varphi. \tag{3.2}$$

This procedure can be applied to advance the semi discrete Navier-Stokes equations (2.17) in time. First an intermediate velocity $\mathbf{u}^*$ is obtained by advancing the velocity from the previous time step without taking ∇p in (2.17) into account.

$$\mathbf{u}^* = \mathbf{u} + \Delta t \left(-\left(\mathbf{u} \cdot \nabla\right) \mathbf{u} + \frac{\mu}{\rho}\nabla^2\mathbf{u} + \mathbf{g} \right) \tag{3.3}$$

The intermediate velocity field is then used as the right hand side of the Poisson equation for the scaled pressure $p^* = p\Delta t$

$$\nabla \cdot \left(\frac{\nabla p^*}{\rho} \right) = \nabla \cdot \mathbf{u}^*. \tag{3.4}$$

Finally the intermediate velocity is made divergence free using the solution of the pressure Poisson equation (3.4)

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\nabla p^*}{\rho}. \tag{3.5}$$

To get an exact discrete projection it is important that ∇p^* in equation (3.4) and (3.5) is discretised in the same way. For a mathematically more rigorous discussion of the projection method, the reader is referred to the book by Chorin and Marsden [13].

3.1.2 Spacial discretisation

The spacial discretisation is done on a staggered grid [30], where the scalar variables (i.e. pressure) are stored at the cell centres and the vector components are stored at the cell faces. Figure 3.1 illustrates the staggered grid. The solid lines indicate the cell boundaries, where the velocity components are stored (the horizontal velocity component is stored on the vertical cell face and vice versa). The grid lines connecting the cell centres are drawn with dashed lines. The boundary of the domain coincidences with the cell boundaries. The fixed grid spacing is Δx in x-direction and Δy in y-direction.

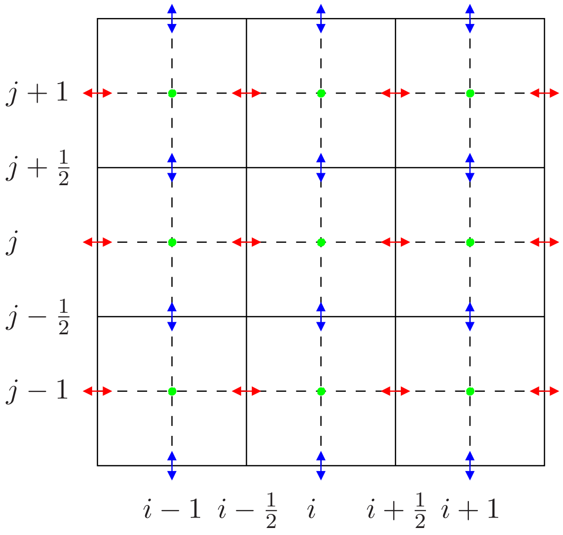

Figure 3.1: Sketch of the staggered grid, scalars are stored at the cell centres (green points) and the vector components are stored at the cell faces (red and blue arrows).

Advective terms

The gradient in the advective terms $(\mathbf{u}\cdot\nabla)\mathbf{u}$ is discretised with a 5th order weighted essentially non-oscillatory (WENO) scheme [38, 25], since the WENO scheme can handle discontinuities in the velocity automatically. In two-dimensions the advective acceleration components read $u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}$ and $u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}$, in x- and y-direction, respectively. First, the velocity components have to be interpolated to the position of the other component. A simple linear interpolation is used, such that the

y-component at the position of the x-component becomes:

$$v_{i-1/2,j} = \frac{v_{i-1,j+1/2} + v_{i,j+1/2} + v_{i-1,j-1/2} + v_{i,j-1/2}}{4}. \tag{3.6}$$

The interpolation of the x-component to the position of the y-component is treated analogously. To form the WENO scheme for $a\frac{\partial\varphi}{\partial x}$ at x_i and y_j, the differences between the neighbouring grid points have to be computed. Depending on the upwind direction we set:

$$\begin{aligned} \Delta v_1 &= \frac{\varphi_{i-5/2} - \varphi_{i-7/2}}{\Delta x}, & \Delta v_2 &= \frac{\varphi_{i-3/2} - \varphi_{i-5/2}}{\Delta x} \\ \Delta v_3 &= \frac{\varphi_{i-1/2} - \varphi_{i-3/2}}{\Delta x}, & \Delta v_4 &= \frac{\varphi_{i+1/2} - \varphi_{i-1/2}}{\Delta x} \\ \Delta v_5 &= \frac{\varphi_{i+3/2} - \varphi_{i+1/2}}{\Delta x}, \end{aligned} \tag{3.7}$$

if $a_{i-1/2} > 0$, or

$$\begin{aligned} \Delta v_1 &= \frac{\varphi_{i+5/2} - \varphi_{i+3/2}}{\Delta x}, & \Delta v_2 &= \frac{\varphi_{i+3/2} - \varphi_{i+1/2}}{\Delta x} \\ \Delta v_3 &= \frac{\varphi_{i+1/2} - \varphi_{i-1/2}}{\Delta x}, & \Delta v_4 &= \frac{\varphi_{i-1/2} - \varphi_{i-3/2}}{\Delta x} \\ \Delta v_5 &= \frac{\varphi_{i-3/2} - \varphi_{i-5/2}}{\Delta x}, \end{aligned} \tag{3.8}$$

in the case of $a_{i-1/2} < 0$. If $a_{i-1/2} = 0$ we do not need to compute the gradient since the convective transport is 0. These differences are used to find an expression for the smoothness of each of the three sub stencils:

$$\begin{aligned} S_1 &= \frac{13}{12}\left(\Delta v_1 - 2\Delta v_2 + \Delta v_3\right)^2 + \frac{1}{4}\left(\Delta v_1 - 4\Delta v_2 + 3\Delta v_3\right)^2 \\ S_2 &= \frac{13}{12}\left(\Delta v_2 - 2\Delta v_3 + \Delta v_4\right)^2 + \frac{1}{4}\left(\Delta v_2 - \Delta v_4\right)^2 \qquad (3.9) \\ S_3 &= \frac{13}{12}\left(\Delta v_3 - 2\Delta v_4 + \Delta v_5\right)^2 + \frac{1}{4}\left(\Delta v_3 - 4\Delta v_4 + \Delta v_5\right)^2. \qquad (3.10) \end{aligned}$$

The smoother a sub stencil the smaller is S. The smoothness indicators are then used to compute a weight for each sub stencil:

$$\begin{aligned} b_1 &= \frac{1}{10}\frac{1}{(\epsilon + S_1)^2}, & w_1 &= \frac{b_1}{b_1 + b_2 + b_3} \\ b_2 &= \frac{6}{10}\frac{1}{(\epsilon + S_2)^2}, & w_2 &= \frac{b_2}{b_1 + b_2 + b_3} \\ b_3 &= \frac{3}{10}\frac{1}{(\epsilon + S_3)^2}, & w_3 &= \frac{b_1}{b_1 + b_2 + b_3}, \end{aligned} \tag{3.11}$$

where ϵ is a small number to avoid a division with 0. In the present work the standard value $\epsilon = 10^{-6}$ was used . Finally the gradient is computed as

$$\begin{aligned} \frac{\partial \varphi}{\partial x}|_{i-1/2} \approx & w_1 \left(\frac{\Delta v_1}{3} - \frac{7\Delta v_2}{6} + \frac{11\Delta v_3}{6} \right) \\ & + w_2 \left(\frac{\Delta v_2}{6} + \frac{5\Delta v_3}{6} + \frac{4\Delta v_4}{6} \right) \\ & + w_3 \left(\frac{\Delta v_3}{3} + \frac{5\Delta v_4}{6} - \frac{\Delta v_5}{6} \right). \end{aligned} \tag{3.12}$$

For smooth flow, the WENO method (3.12) is 5th order accurate. For non smooth flow, the accuracy is reduced to at least third order, depending on the relative position of the discontinuity in relation to the stencil.

Viscous terms

The viscous terms $\mu\nabla^2\mathbf{u}$ in the absence of an interface are computed with a standard second order 5 point stencil. The x-component of the viscous term reads

$$\begin{aligned} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)|_{i-1/2,j} \approx & \frac{u_{i-3/2,j} - 2u_{i-1/2,j} + u_{i+1/2,j}}{\Delta x^2} \\ & + \frac{u_{i-1/2,j+1} - 2u_{i-1/2,j} + u_{i-1/2,j-1}}{\Delta y^2}, \end{aligned} \tag{3.13}$$

while the y-component is treated in the same way.

Pressure Poisson equation

The divergence of the intermediate velocity field on the right hand side of the pressure Poisson equation (3.4) is approximated by

$$\nabla \mathbf{u}^*|_{i,j} \approx \frac{u^*_{i+1/2,j} - u^*_{i-1/2,j}}{\Delta x} + \frac{v^*_{i,j+1/2} - u^*_{i,j-1/2}}{\Delta y}. \tag{3.14}$$

The Laplace operator for the pressure away form interfaces is the same as the one used for the viscous terms (3.13). The resulting linear system for the pressure is solved using a direct block tridiagonal solver. Finally, the gradient of the pressure is computed as

$$\frac{\partial p^*}{\partial x}|_{i-1/2,j} \approx \frac{p^*_{i,j} - p^*_{i-1,j}}{\Delta x} \tag{3.15}$$

$$\frac{\partial p^*}{\partial y}|_{i,j-1/2} \approx \frac{p^*_{i,j} - p^*_{i,j-1}}{\Delta y}. \tag{3.16}$$

3.2 Level set method

There are several different methods to keep track of the interface position in multiphase flows. A popular method is the level set method (LSM) [60, 51], in which the interface is defined as the zero contour line of a scalar function ϕ. Typically ϕ is the signed distance function from the interface. The signed distance function exists and is continuous in the entire computation domain. The signed distance function is advected with the local fluid velocity using the advection equation

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0. \tag{3.17}$$

For the gradient of ϕ the same 5th order WENO scheme as described in section 3.1.2 is used. The scalar function ϕ is stored at the cell centres. Thus the velocity components have to be interpolated to the cell centres which is done by simple linear interpolation, i.e. $u_{i,j} = 1/2\left(u_{i-1/2,j} + u_{i+1/2,j}\right)$ and $v_{i,j} = 1/2\left(v_{i,j-1/2} + v_{i,j+1/2}\right)$.

Since the discretisation of the advection equation is not exact and since the fluid velocity $\mathbf{u}$ is not constant throughout the domain, ϕ loses its signed distance property over time. To keep ϕ close to a signed distance function, it has to be reinitialised with a regular interval. This is achieved by solving the following equation to steady state

$$\frac{\partial \phi}{\partial \tau} = -S(\phi)\left(|\nabla \phi| - 1\right), \tag{3.18}$$

where τ is an artificial time. The reinitialisation of ϕ is implemented with the help of the Level Set Method Library (LSMLIB) [14]. Again the 5th order WENO scheme is used to approximate the gradient of ϕ, where Godunov's Scheme [51, see chapter 5.3.3] is used to determine the upwind direction. The smeared sign function is computed as $S(\phi) = \frac{\phi_0}{\sqrt{\phi_0^2 + \max(\Delta x, \Delta y)^2}}$, where ϕ_0 is ϕ at the beginning of the reinitialisation.

The interface normal and curvature can be obtained directly from the signed distance function.

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|} \tag{3.19}$$

$$\kappa = -\nabla \cdot \mathbf{n} \tag{3.20}$$

The normal is always pointing in the direction of the fluid which occupies the region with a positive ϕ. The sign of the curvature is used to identify whether the interface is convex or concave seen from the fluid with negative ϕ. Thus it is used to decide whether the pressure is de- or increasing over the interface. The normal (3.19) is approximated by a conventional second order finite difference scheme. The expression for the curvature (3.20) can be rewritten in Cartesian coordinates:

$$\kappa = -\frac{\left(\frac{\partial \phi}{\partial x}\right)^2 \frac{\partial^2 \phi}{\partial y^2} - 2\frac{\partial \phi}{\partial x}\frac{\partial \phi}{\partial y}\frac{\partial^2 \phi}{\partial x \partial y} + \left(\frac{\partial \phi}{\partial y}\right)^2 \frac{\partial^2 \phi}{\partial x^2}}{\left(\left(\frac{\partial \phi}{\partial x}\right)^2 + \left(\frac{\partial \phi}{\partial y}\right)^2\right)^{3/2}}, \tag{3.21}$$

where the derivatives are approximated by second order finite differences as well.

3.2.1 Constrained reinitialisation

It can be observed that the interface is displaced during the reinitialisation see [58] for a simple 1-dimensional example. Russo and Smereka

[58] noted that the reinitialisation equation (3.18) is hyperbolic and the characteristics are pointing away from the interface. This means that at the interface the characteristics are changing their direction. For grid points adjacent to the interface the upwind direction is towards the interface. But since the interface in general is not located at grid points, an upwind discretisation at such a grid point would include points on the opposite side of the interface, and there the characteristics point in the opposite direction. Thus at grid points adjacent to the interface there is no proper upwind stencil. Russo and Smereka [58] proposed to fix this problem by interpolating the interface position and then use the interface instead of the next grid point to form an upwind stencil. Hartmann et al. [31] introduced the constrained reinitialisation (CR) which forces the interface to keep its position during the reinitialisation. This method was then reformulated such that the forcing term appears as a source term in equation (3.18) [32] which makes it possible to implement the constrained reinitialisation without changing the discretisation stencils. In the current work the HCR-1 scheme from [32] was used to reduce the spurious displacement of the interface.

The idea of the HCR-1 scheme is to add a source term to the residual of the differential equation, which minimises the displacement of the intersection points between the zero contour of ϕ and the grid lines in a least squares sense. If two grid points $\mathbf{x}_{i-1,j}$ and $\mathbf{x}_{i,j}$ which are located on opposite sides of the interface, the condition that the linear interpolation of the intersection point between the zero contour line and the grid line between those two points does not move during reinitialisation can be reduced to $\frac{\tilde{\phi}_{i,j}}{\tilde{\phi}_{i-1,j}} = \frac{\phi_{i,j}}{\phi_{i-1,j}}$, where $\tilde{\phi}$ and ϕ are the signed distance functions before and after reinitialisation, respectively. In general a grid point can have several neighbours which are on the opposite side of an interface. The previous condition cannot be fulfilled for all involved neighbours, since the problem is over determined. Let $S_{i,j}$ be the set of all neighbouring grid points of $\mathbf{x}_{i,j}$ which are on the opposite side of the interface, and $M_{i,j}$ the number of grid points in $S_{i,j}$. Further we denote an arbitrary point in $S_{i,j}$ by $\mathbf{x}_{(i,j)_\alpha}$ such that $\alpha = 1 ... M_{i,j}$. In the constrained reinitialisation HCR-1 [32] the following least squares function is minimised.

$$L_{i,j} = \sum_{\alpha=1}^{M_{i,j}} \delta_{i,j}^{\alpha} \left(\phi_{i,j} - \phi_{(i,j)_\alpha} \cdot r_{(i,j)_\alpha}^{i,j} \right)^2 , \tag{3.22}$$

$$r_{(i,j)_\alpha}^{i,j} = \frac{\tilde{\phi}_{i,j}}{\tilde{\phi}_{i-1,j}} . \tag{3.23}$$

Hartmann et al. [32] chose the weights $\delta_{i,j}^{\alpha} = 1$. If $L_{i,j}$ is differentiated with respect to $\phi_{i,j}$ and this is set equal to zero, we get the target value for the distorted signed distance function such that it minimises the displacement of the interface:

$$T_{i,j} = \frac{\sum_{\alpha=1}^{M_{i,j}} \delta_{i,j}^{\alpha} \left(\phi_{(i,j)_\alpha} \cdot r_{(i,j)_\alpha}^{i,j} \right)}{\sum_{\alpha=1}^{M_{i,j}} \delta_{i,j}^{\alpha}} . \tag{3.24}$$

Finally the HCR-1 correction term at the nth reinitialisation time step is formulated as the difference between $T_{i,j}^n$ and $\phi_{i,j}^n$:

$$F_{i,j}^n = \frac{\beta_{i,j}}{\min(\Delta x, \Delta y)} \left(\phi_{i,j}^n - T_{i,j}^n \right) . \tag{3.25}$$

Here $\beta_{i,j}$ is a coefficient which distributes the correction between neighbouring grid points. For consistency the sum $\beta_{i,j} + \beta_{(i,j)_\alpha}$ must always be equal to 1. Setting $\beta_{i,j} = 0.5$ corresponds to do half of the correction on either side of the interface. Since the forcing leads to an instability when one of the points of the set $S_{i,j}$ changes its sign during reinitialisation, the forcing is not applied at the grid points in this particular set $S_{i,j}$. The correction term $F_{i,j}^n$ is added to the discretised right hand side of equation (3.18).

3.3 Ghost fluid method

To extend the single phase Navier-Stokes solver from section 3.1 it is not sufficient to know the interface position. As it was demonstrated in section 2.2, we have discontinuities in the solution. The ordinary stencils cannot handle those discontinuities and will lose their accuracy if jumps are present. The ghost fluid method (GFM) was originally developed to handle the discontinuities in compressible flow computations [25]. Later it was generalised. Liu et al. [47] used the GFM to solve Poisson equations with immersed boundaries and known jump conditions. The stencils formulated in [47] were then used to implement the interface conditions in incompressible two-phase flow [40]. The idea of the GFM is to introduce ghost cells on either side of the interface. The values in those ghost cells can be computed knowing the interface conditions and the values of the real fluid at this point. This explains the name of the method despite the fact that for the formulation used here there is no need to have multiple values for the variables at one grid point.

Here we will only sketch the GFM for a one-dimensional problem, since it can be implemented dimension for dimension. Consider the Poisson equation

$$\frac{\partial}{\partial x}\left(\beta\frac{\partial\varphi}{\partial x}\right) = f, \tag{3.26}$$

where φ and f are scalar functions. The coefficient β can take a different constant value on each side of an interface. Further assume that there is an interface at x_Γ with the interface conditions

$$[\![\varphi]\!] = a_\Gamma, \qquad \left[\!\!\left[\beta\frac{\partial\varphi}{\partial x}\right]\!\!\right] = b_\Gamma. \tag{3.27}$$

The interface location can be described by a level set function ϕ. Suppose now that the level set function for two neighbouring grid points are $\phi_k \leq 0$ and $\phi_{k+1} > 0$, which means that x_Γ is located between x_k and x_{k+1}. The sign of the level set function is used to decide what the value of β is. To the left of the interface we have β^- and to the right β^+. The situation is sketched in Figure 3.2. If $b_\Gamma = 0$, the first derivative of φ at $x_{k-1/2}$ and $x_{k+1/2}$ can be approximated as

$$\frac{\partial\varphi}{\partial x}|_{k-1/2} \approx \frac{\varphi_k - \varphi_{k-1}}{\Delta x} \tag{3.28}$$

$$\frac{\partial\varphi}{\partial x}|_{k+1/2} \approx \frac{(\varphi_{k+1} - a_\Gamma) - \varphi_k}{\Delta x}. \tag{3.29}$$

Since x_{k+1} is on the opposite side of the interface a was subtracted from the value of φ_{k+1}. For this would be the value, which φ would have, if x_{k+1} were to belong to the left fluid domain, cf 3.2. Figure.

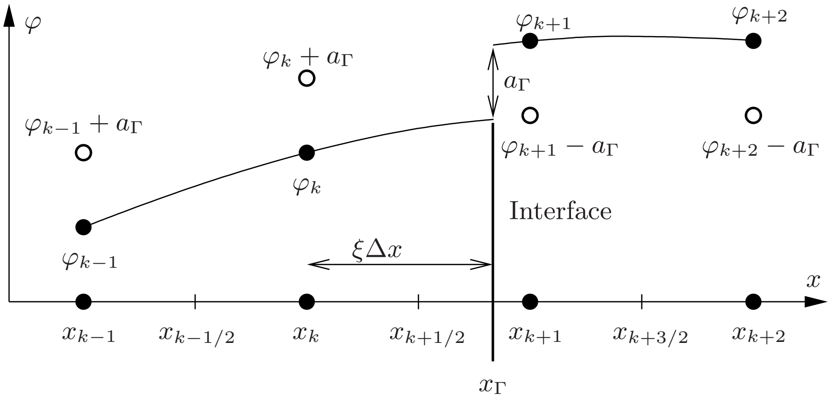

Figure 3.2: Sketch of the interface position between two grid points.

To correct for the discontinuity in the first derivative (i.e. $b_\Gamma \neq 0$), the the first derivative on both sides of the interface is approximated. The difference of those two approximations must equal the second interface condition

$$\beta^+ \frac{(\varphi_{k+1} - a_\Gamma) - \varphi_\Gamma}{(1-\xi)\Delta x} - \beta^- \frac{\varphi_\Gamma - \varphi_k}{\xi \Delta x} = b_\Gamma, \tag{3.30}$$

where $\xi \Delta x$ is the distance between the interface and the grid point x_k. This expression can be solved for the solution at the interface

$$\varphi_\Gamma = \frac{\hat{\beta}(1-\xi)}{\beta^+}\varphi_k + \frac{\hat{\beta}\xi}{\beta^-}(\varphi_{k+1} - a_\Gamma) - \frac{\hat{\beta} b_\Gamma \Delta x \xi (1-\xi)}{\beta^+ \beta^-}, \tag{3.31}$$

where

$$\hat{\beta} = \frac{\beta^+ \beta^-}{\beta^+ \xi + \beta^-(1-\xi)}. \tag{3.32}$$

Using φ_Γ we can approximate the first derivative of φ at $x_{k+1/2}$

$$\frac{\partial \varphi}{\partial x}|_{k+1/2} \approx \frac{\varphi_\Gamma - \varphi_k}{\xi \Delta x}. \tag{3.33}$$

Finally the approximation for the Laplace operator at x_k is obtained

$$\frac{\partial}{\partial x}\left(\beta\frac{\partial\varphi}{\partial x}\right)|_{x_k} \approx \left(\frac{\hat{\beta}\left(\varphi_{k+1}-\varphi_k\right)}{\Delta x^2} - \frac{\beta^-\left(\varphi_k-\varphi_{k-1}\right)}{\Delta x^2}\right) - \frac{\hat{\beta}a_\Gamma}{\Delta x^2} - \frac{\hat{\beta}b_\Gamma(1-\xi)}{\beta^+\Delta x}. \tag{3.34}$$

Using the same approach, the approximation for the Laplace operator at x_{k+1} becomes

$$\frac{\partial}{\partial x}\left(\beta\frac{\partial\varphi}{\partial x}\right)|_{x_{k+1}} \approx \left(\frac{\beta^+\left(\varphi_{k+2}-\varphi_{k+1}\right)}{\Delta x^2} - \frac{\hat{\beta}\left(\varphi_{k+1}-\varphi_k\right)}{\Delta x^2}\right) + \frac{\hat{\beta}a_\Gamma}{\Delta x^2} - \frac{\hat{\beta}b_\Gamma\xi}{\beta^-\Delta x}. \tag{3.35}$$

In the case where $\varphi_k > 0$ and $\varphi_{k+1} \leq 0$ we have to exchange β^+ with β^- and vice versa as well as the signs in front of both of the correction terms have to be inverted, since the jump is defined as $[\![\varphi]\!] = \varphi^+ - \varphi^-$. If there is no interface (i.e. $a_\Gamma = 0$, $b_\Gamma = 0$ and $\beta^- = \beta^+ = \hat{\beta} = \beta$) the ordinary second order finite difference stencil (3.13) is retrieved. If we want to solve the Poisson equation (3.26) the correction terms containing a_Γ and b_Γ can be moved to the right hand side such that the linear system is still symmetric and negative definite, which is desirable if an iterative solver is to be used. Interpolating the position where the level set function is zero gives $\xi = \frac{|\phi_k|}{|\phi_k|+|\phi_{k+1}|}$.

In two-phase flow, we have both discontinuities in the pressure and in the first derivative of the velocity at the interface. Therefore at the interface the discretisation for the viscous term (3.13) has to be replaced by the stencil obtained form the GFM. For the viscous terms $a_\Gamma = 0$ because of (2.24), b_Γ can be computed using equation (2.38) and $\beta = \mu$. As the interface conditions are continuous, they are computed at the grid nodes (see Figure 3.1) and then interpolated to the interface position where they are required. Kang et al. [40] choose to first interpolate the velocity components to the cell centres and then standard central differences are used to approximate the gradients of the velocity at the cell centres. Here a different strategy is adopted, where the derivatives u_x and v_y are directly computed as the difference approximations of the velocities at the cell faces, i.e., $\frac{\partial u}{\partial x}|_{i,j} = \left(u_{i+1/2,j} - u_{i-1/2,j}\right)/\Delta x$, and the other two derivatives namely $\frac{\partial u}{\partial y}$ and $\frac{\partial v}{\partial x}$ are approximated at the cell corners, i.e., $\frac{\partial u}{\partial y}|_{i+1/2,j+1/2} = \left(u_{i+1/2,j+1} - u_{i+1/2,j}\right)/\Delta y$ and those two derivatives are subsequently interpolated to the cell centres by averaging over the four corner values. Both methods lead to the same stencil for u_y and v_x, but for u_x and v_y the present direct approximation results in a smaller stencil with the same order of accuracy. In addition we point out that u_x and v_y at the cell centres and $\frac{\partial u}{\partial y}$ and $\frac{\partial v}{\partial x}$ at the cell corners have to be computed anyway to approximate the viscous terms, cf. equations (3.13). The derivatives of the velocities are then multiplied by the appropriate components of the normal and tangential vectors, which are located at the grid centres as well. Finally the values of the interface corrections at the grid centres are interpolated to the location of the interface using the level set function $b_\Gamma = \frac{b_k|\phi_{k+1}|+b_{k+1}|\phi_k|}{|\phi_k|+|\phi_{k+1}|}$.

The discrete Laplace operator for the pressure at the interface is also replaced by the GFM. But here a_Γ is given by equation (2.33), $b_\Gamma = 0$ [see 40, section 3.7 for the justification] and $\beta = 1/\rho$. Again a is computed at the cell centres and then interpolated to the required interface position using the level set function $a_\Gamma = \frac{a_k|\phi_{k+1}|+a_{k+1}|\phi_k|}{|\phi_k|+|\phi_{k+1}|}$. It has to be remembered that we are solving for the pressure scaled by the time step p^* and therefore also the interface condition (2.33) has to be multiplied by Δt.

Finally the discrete gradient of the pressure correction has to be replaced by the GFM at the interface. Setting $b = 0$ the x-component of the gradient multiplied by β becomes

$$\beta\frac{\partial\varphi}{\partial x}|_{k+1/2} = \hat{\beta}\frac{(\varphi_{k+1} - a_\Gamma) - \varphi_k}{\Delta x}, \tag{3.36}$$

if $\varphi_k \leq 0$ and $\varphi_{k+1} > 0$, or otherwise

$$\beta\frac{\partial\varphi}{\partial x}|_{k+1/2} = \hat{\beta}\frac{(\varphi_{k+1} + a_\Gamma) - \varphi_k}{\Delta x}, \tag{3.37}$$

where $\hat{\beta}$ is given in equation (3.32).

3.4 Contact point treatment

Spelt [63] introduced an explicitly tracked contact point moving along the solid boundary as illustrated in Figure 3.3. The contact point is tracked explicitly using the ordinary differential equation

$$\frac{\mathrm{d}x_{\mathrm{CP}}}{\mathrm{d}t} = u_{\mathrm{CP}} = f(\theta_{\mathrm{CP}}), \tag{3.38}$$

where $f(\theta_{\mathrm{CP}})$ is a function describing the dependency of the contact point velocity on the contact angle θ_{CP}. The choice of f is important and should be done carefully to obtain reliable results. Different approaches to model the dependency of the contact point velocity on the contact angle are possible, including the use of empirical data or results from micro scale simulations around the interface. The method presented here is not restricted to contact point velocities which are a function of the contact angle. In principle $f(\theta)$ can be replaced by any relation. Immediately after the advection of ϕ the contact point is updated using the velocity computed from $f(\theta)$ at the last time step.

$$x_{\mathrm{CP}}^{n+1} = x_{\mathrm{CP}}^{n} + \Delta t \cdot f(\theta_{\mathrm{CP}}^{n}) \tag{3.39}$$

Higher order in time is achieved by using the explicit Euler method (3.39) for (3.38) in each stage of an explicit Runge-Kutta method as discussed in section 3.5.

3.4.1 Boundary conditions for reinitialisation

The coupling between the zero level set of the distance function ϕ and the explicitly tracked contact point has to be ensured. This coupling can be enforced by the use of appropriate boundary conditions for ϕ during the reinitialisation. Since the reinitialisation equation (3.18) is hyperbolic and the characteristics are straight lines, which are pointing perpendicular away from the interface. The position of the interface decides where a boundary condition for the reinitialisation is required. If the interface is closed and lies completely inside the computation domain, no boundary conditions are necessary. In areas where no boundary condition is required the ghost points for the signed distance function can simply be filled with a linear extrapolation from the fluid points. The linear

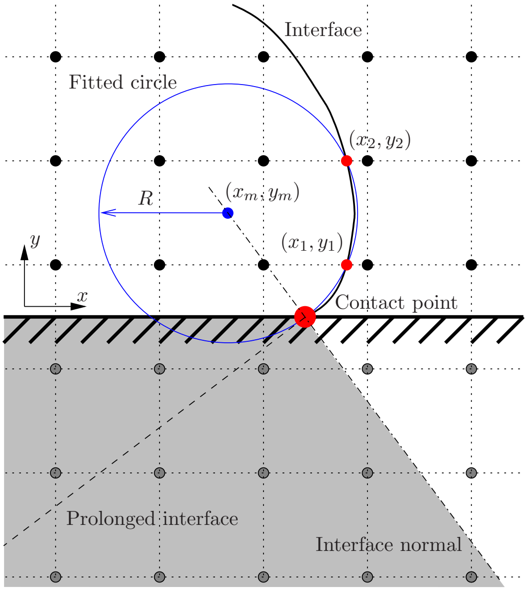

Figure 3.3: Sketch of the explicitly tracked contact point.

extrapolated values in the ghost points will force the WENO scheme to select stencils which are biased towards the direction where the characteristics are coming from, i.e. the inside of the computation domain. At locations where the interface is crossing the domain boundary with an angle not equal to 90°, i.e. at a contact point we will have characteristics entering the computation domain. Incoming characteristics in a hyperbolic equation require that boundary conditions are set. To fill the ghost points in areas where a boundary condition is required, the interface is prolonged as a straight line from the contact point with a slope given by θ_{CP} [63]. The idea of letting the interface cross the wall to compute a boundary condition for the signed distance function or the VOF colour function is also used by Liu et al. [46] and Renardy et al. [57]. A coordinate system where the x-axis is parallel to the wall and the y-axis is pointing into the fluid domain (see Figure 3.3) is introduced. The slope of the prolonged interface follows the equation

$$y_I = S_b \tan(\theta_{\mathrm{CP}}) \, (x - x_{\mathrm{CP}}) \, , \tag{3.40}$$

where S_b is the sign of the level set function to the left of the interface. S_b is introduced since the contact angle θ_{CP} is always measured in the fluid with negative level set function. We use $a = S_b \tan(\theta_{\mathrm{CP}})$ to simplify the notation. The interface normal at the contact point is given as

$$y_N = -\frac{1}{a}\left(x - x_{\mathrm{CP}}\right). \tag{3.41}$$

For ghost points at (x_i, y_j), where we assume that the first grid point in the fluid domain has the index $j = 1$, the distance to the prolonged interface is

$$d_{i,j} = \sqrt{\frac{\left(x_i - x_{\mathrm{CP}}\right)^2}{a^2} + \left(x_i - x_{\mathrm{CP}} + \frac{x_{\mathrm{CP}} - x_i + a y_j}{a^2 + 1}\right)^2}. \tag{3.42}$$

To get the signed distance to the prolonged interface $\tilde{\phi}_{i,j} = s d_{i,j}$ is set, if $y_j > a(x_i - x_{\mathrm{CP}})$ (i.e. in the ghost points above the prolonged interface) or $\tilde{\phi}_{i,j} = -s d_{i,j}$ is set otherwise. Here $s = S_b \mathrm{sign}(a)$ is the sign of ϕ above the contact point. To obey the characteristics it is important that only the ghost points in the shaded region in Figure 3.3 get the signed distance to the prolonged interface $\tilde{\phi}$. All the other ghost points should just get linear extrapolated values as described above. The same holds true for areas where there is a real interface in the fluid domain which is closer to the wall than the prolonged interface.

As it was mentioned by Spelt [63], just taking $\tilde{\phi}$ or the linearly extrapolated value of ϕ, whichever has the smaller absolute value can lead to the erroneous selection of the extrapolated value. This can effectively be avoided by replacing the ordinary linear extrapolation by a signed linear extrapolation [48]

$$\hat{\phi}_{i,j} = \phi_{i,1} - \mathrm{sign}\left(\phi_{i,\mathrm{wall}}\right)\left|\phi_{i,2} - \phi_{i,1}\right|(j-1) \qquad j = -2\ldots 0, \tag{3.43}$$

where $\mathrm{sign}\left(\phi_{i,\mathrm{wall}}\right)$ is the sign of ϕ at the wall at the position of the extrapolation. It must be determined using the contact point position. Because if $\phi_{i,\mathrm{wall}}$ is determined by extrapolation from the fluid domain, it is the extrapolation error, which could cause the erroneous selection of the extrapolated value as mentioned above, and one would lose the advantage of the signed linear extrapolation.The signed extrapolation ensures that the absolute value of the distance function in the ghost point is always growing as the distance to the wall is increased. This leads to two effects, first it is ensured that no spurious interfaces are introduced through the boundary condition (which is the reason it was introduced by Miller [48]) and secondly it ensures that the the distance to the prolonged interface is selected where there are incoming characteristics.

In the end the ghost points are filled with either $\hat{\phi}$ or $\tilde{\phi}$, whichever has the smaller absolute value.

3.4.2 Curvature and interface angle at the wall

At the wall the curvature κ and the contact angle cannot be computed with central differences like it is done in the interior (cf. section 3.2), since the stencil for the first grid point inside the fluid domain would contain points which are located inside the wall. As discussed above, the interface is assumed to continue as a straight line into the wall and the level set values at the ghost points are set as signed distances to that line. As a result the curvature at the first grid point inside the fluid domain would be compromised if computed by central differences. Although the accuracy problem could be solved with one-sided stencils, the curvature computation with finite differences would not be directly dependent on the contact point position. Thus there would not be a direct coupling between the contact point position x_{CP} and the flow field determined by the Navier-Stokes equations (2.17).

Instead a circle is fitted through the contact point x_{CP} and the intersection points of the interface with the first two grid lines parallel to the wall. The fitted circle is illustrated in Figure 3.3, where the intersection points (x_1, y_1) and (x_2, y_2), and the contact point are highlighted as red dots. The centre of the circle is given as

$$\begin{aligned} x_m &= \frac{-m_a m_b y_2 + m_b\left(x_{\mathrm{CP}} - x_1\right) - m_a\left(x_1 + x_2\right)}{2\left(m_b - m_a\right)} \\ y_m &= \frac{\left(x_{\mathrm{CP}} - x_2\right) + m_a y_1 - m_b\left(y_1 + y_2\right)}{2\left(m_a - m_b\right)}, \end{aligned} \tag{3.44}$$

where $m_a = y_1 / \left(x_1 - x_{\mathrm{CP}}\right)$ and $m_b = \left(y_2 - y_1\right) / \left(x_2 -_x 1\right)$. The curvature at the first grid point is then given as the inverse of the radius from the fitted circle

$$\kappa = \frac{1}{\sqrt{\left(x_m - x_{\mathrm{CP}}\right)^2 + y_m^2}}. \tag{3.45}$$

The sign of the curvature is adjusted to make it consistent with the curvature in equation (3.20) by multiplying κ with $\operatorname{sign}\left(x_{\mathrm{CP}} - x_m\right) * S_b$. The smaller angle between the wall and the fitted circle is

$$\hat{\theta} = \sin^{-1}\left(\frac{\left|x_{\mathrm{CP}} - x_m\right|}{\sqrt{\left(x_m - x_{\mathrm{CP}}\right)^2 + y_m^2}}\right). \tag{3.46}$$

Depending on the position of the circle centre (x_m, y_m) it can be decided if $\hat{\theta}$ is the contact angle in the fluid with negative ϕ. If $S_b < 0$ and $\left(x_{\mathrm{CP}} - x_m\right) y_m > 0$ or if $S_b > 0$ and $\left(x_{\mathrm{CP}} - x_m\right) y_m \leq 0$ the contact angle in the negative fluid is $\theta_{\mathrm{CP}} = \pi - \hat{\theta}$, in all other cases we have $\theta_{\mathrm{CP}} = \hat{\theta}$.

During the reinitialisation the interface suffers from spurious displacement. Therefore the reinitialisation will change the measured contact angle and curvature, since they depend on the first two interpolated intersection points. The spurious displacement and therefore the change in the contact angle and curvature can be efficiently reduced using a constrained reinitialisation [32] as discussed in section 3.2.1.

3.4.3 Interpolation of the Intersection points

Since the curvature adjacent to the wall is computed from the fitted circle, the intersection points through which the circle is fitted need to be approximated with sufficient accuracy. The curvature of the fitted circle is two orders less accurate than the approximation of the points on the circle, because the curvature is a function of the second derivative of the curve fitted through these points. To get an order of accuracy for the curvature at the wall which is consistent with the accuracy of the curvature in the interior, the intersection points have to be interpolated form ϕ with a fourth order accurate scheme. However, in our numerical experiments we had cases where the curvature did not converge despite using cubic interpolation for the intersection points.

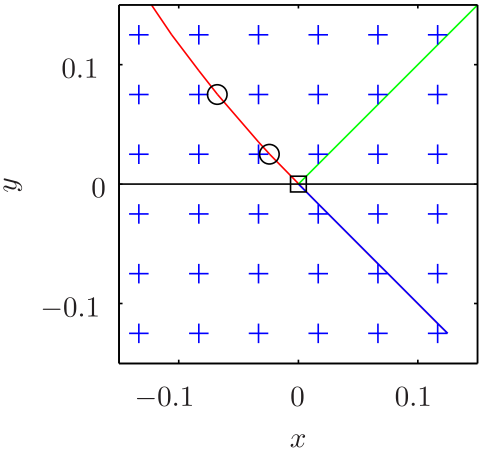

Figure 3.4: A circular interface (red line) intersects the wall (black line), and is prolonged as a straight line (blue) into the ghost point region. The square is the location of the contact point and the circles indicate the positions where the interface intersects the grid lines.

This is illustrated with a simple example. A circle with unit radius is placed such that it intersects the wall with an angle of 45°. The interface is prolonged as a straight line. The level set function for this situation can be computed analytically. The example is illustrated in Figure 3.4. The position of the intersection points is now computed from ϕ using linear, quadratic and cubic interpolation. The stencils for the linear and cubic interpolation are centred around the intersection points and the quadratic stencil uses one grid point to the right and two points to the left of the interface. The interpolation error for the three different interpolation methods using n grid points per radius are plotted in Figure 3.5. Both the linear and cubic interpolation converge only with second order, while the quadratic interpolation exhibits its full convergence order. As a consequence the curvature error, which is shown in Figure 3.6, does not converge for the linear and cubic interpolation. The curvature error for the quadratic interpolation converges linearly as expected. The reason for this lack of convergence is a discontinuity in the second derivatives of ϕ. At the contact point, the circle is prolonged as a straight line, meaning that the curvature of the interface jumps from a finite value to zero. The characteristics, which are pointing perpendicularly away form the interface, intersect along a line which coincides with the interface normal at the contact point. This is leading to the mentioned discontinuity in the second derivatives of

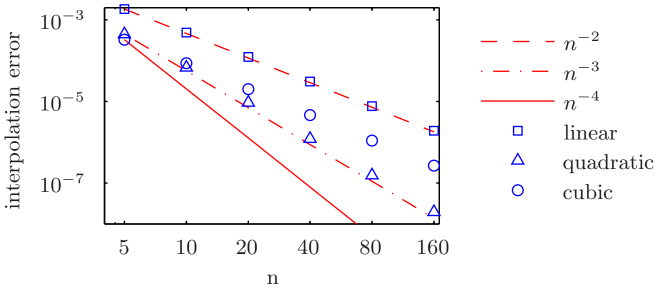

Figure 3.5: Interpolation error of x_1 using standard interpolation stencils.

ϕ. As it can be seen in Figure 3.4 a cubic central interpolation stencil for x_1 crosses the interface normal and therefore the discontinuity. Hence, the convergence order is reduced to two. On the other hand the quadratic interpolation stencil is chosen such that it does not cross the discontinuity and we obtain the full order.

A possible remedy to this problem is using an ENO type interpolation, where additional points in the stencil are chosen based on the smoothness of ϕ. Another possibility is to adapt the position of the interpolation stencil such that is as close to a symmetric stencil as possible without crossing the interface normal. The experiment above was repeated using those two approaches, and the results are presented in Figures 3.7 and 3.8. The ENO interpolation prefers the symmetric stencil despite the discontinuity and therefore the curvature computed with this method does not converge. The adaptive stencil is working as intended, and a curvature which is converging with second order is obtained.

Since the intersection point interpolation is used to fit the circle, the current contact angle is not available at the new time step. In all the presented examples the interpolation stencil is chosen based on the contact angle of the old time step. However, after the fitting of the circle and the computation of the new contact angle at the new time step, one could check, whether the applied interpolation stencil is indeed the optimal one. One could check whether the stencil crosses the interface normal using the newly approximated contact angle. One could even check, whether a more symmetric stencil would be possible.

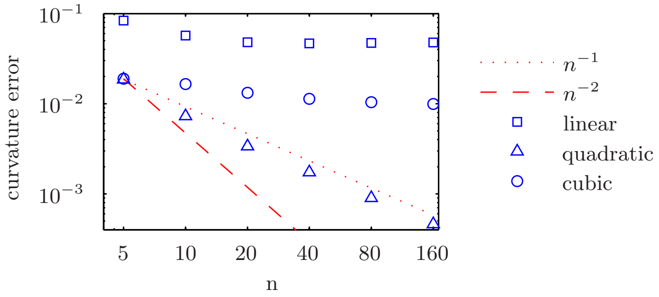

Figure 3.6: Error of the curvature at the contact point, x_1 is interpolated using standard interpolation stencils.

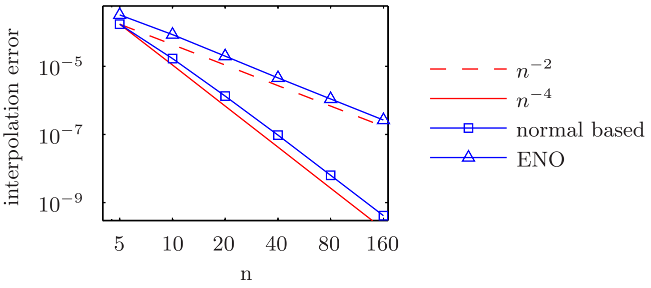

Figure 3.7: Interpolation error of x_1 using interpolations which should omit the discontinuity.

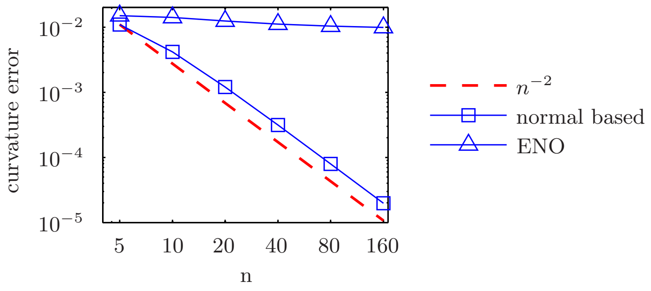

Figure 3.8: Error of the curvature at the contact point, x_1 is interpolated using interpolations which should omit the discontinuity.

3.5 Temporal discretisation

For the temporal discretisation of the semidiscrete equations the 3rd order total variation diminishing (TVD) Runge-Kutta method [62] is employed. It can be written as a weighted average of three stages:

$$\mathbf{U}^{n+1} = \frac{1}{3}\mathbf{U}^n + \frac{2}{3}E\left(\frac{3}{4}\mathbf{U}^n + \frac{1}{4}E\left(E\left(\mathbf{U}^n\right)\right)\right), \tag{3.47}$$

where $E(\mathbf{U}^n) = \mathbf{U}^n + \Delta t R(\mathbf{U}^n)$ denotes one stage which is identical to a forward Euler time step and $\mathbf{U}$ is the solution vector containing the velocity $\mathbf{u}$, the signed distance function ϕ and the contact point position x_{CP}. $R(\mathbf{U}^n)$ is the right hand side of the semidiscrete equations $-(\mathbf{u}\cdot\nabla)\mathbf{u} + \frac{\mu}{\rho}\nabla^2\mathbf{u} + g$, $-\mathbf{u}\cdot\nabla\phi$ and (3.38). The contact angle θ_{CP} and the curvature at the wall have to be recomputed with the procedure presented in section 3.4.2 after the evaluation of the weighted average of $\mathbf{U}$ from the different stages. One stage $E()$ consists of the following steps:

1. Computation of the viscous terms $\mu\nabla^2\mathbf{u}$ using ϕ^n and $\mathbf{n}^n$ to compute the jump conditions (2.38).
2. Advection (3.17) of ϕ.
3. Advection of the contact point (3.39).
4. Reinitialisation (3.18) of ϕ, after every pseudo time step of the reinitialisation θ_{CP} is updated and the boundary conditions are set according to sections 3.4.2 and 3.4.1, respectively. Note that reinitialisation is not required at every time step.
5. Divide the viscous terms by ρ, where ϕ^{n+1} is used to determine the value of ρ.
6. Compute surface normals $\mathbf{n}^{n+1}$ and curvature κ^{n+1} in the interior.
7. Compute the intermediate velocity field $\mathbf{u}^*$ (3.3)
8. Solve the Poisson equation (3.4) and correct the velocity field (3.5), where ϕ^{n+1} is used to compute $\hat{\beta}$ and κ, while the jumps originating form the viscous stresses (i.e. the first term on the right hand side of equation (2.33)) is computed as in step 1.

Chapter 4

Results

In this chapter the simulations of five different two-phase flows are presented. The first test case considers damped surface waves and does not contain any contact points. It is used to assess different implementation details of the two-phase solver. The first example containing contact points is the test case of a capillary rise, where the stationary interface shape is compared to the exact solution. The results for those two test cases are not contained in the attached articles. The remaining examples, i.e. capillary and gravity driven channel flow and an advancing droplet, are used to assess the numerical method for moving contact points. They are taken from the attached articles [e] and [f].

4.1 Damped surface waves

Viscous damping of surface waves is used to verify that the implementation of the two-phase solver, which was presented in chapter 3, is working properly. In addition this case is used to assess a conservative discretisation of the advective terms and a different approximation of the jump conditions.

The problem consists of two fluids which are separated by an interface. Initially the interface is perturbed with a small periodic deformation, such that the horizontal position of the interface is given as

$$y_I(x) = y_0 + A_0 \cos\left(2\pi \frac{x}{\lambda}\right), \tag{4.1}$$

where y_0 is the position of the undisturbed horizontal interface, A_0 is the initial amplitude and $\lambda = 2\pi$ is the wavelength of the perturbation. Due to surface tension σ at the interface, the disturbed interface will lead to pressure jumps across the interface, which in turn will create a nonzero velocity field. The system is stable in the sense that it will return to its undisturbed configuration, i.e. a horizontal interface. As viscous and surface tension forces interact, the amplitude $A(t)$ of the perturbation will decrease over time. An analytical solution for the linearised problem was presented by Prosperetti [52].

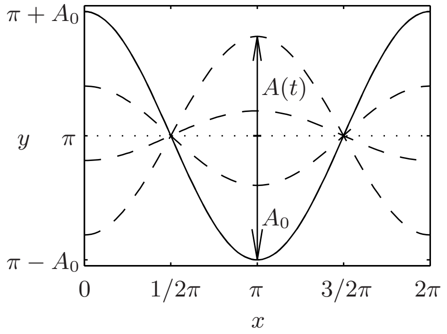

Figure 4.1: Sketch of the geometry of the damped surface wave, the solid line represents the initial interface position and the dashed lines are the interface positions at some arbitrary instants of time with $t > 0$.

We use non-dimensional variables here. The wavelength of the perturbation is $\lambda = 2\pi$ and the initial amplitude is set to $A_0 = 0.01\lambda$. According to the analytical solution the oscillation frequency ω_0 is given by $\omega_0^2 = \left(\frac{2\pi}{\lambda}\right)^3 \frac{\sigma}{\rho^+ + \rho^-}$. All solutions are obtained in a domain of size $\lambda \times \lambda$ where $y_0 = \lambda/2$. The boundary conditions parallel to the interface are symmetric (or walls with complete slip), and perpendicular to the interface periodicity is enforced. The densities of the fluids are $\rho^- = 1000$ and $\rho^+ = 1$, the surface tension is $\sigma = 2$ and the kinematic viscosities are set to $\nu = 0.006472$. For all reported grid resolutions the time step was set to $\Delta t = 0.01$ and the simulation was run until $t_e \omega_0 = 20$

The time evolution of the interface amplitude is plotted in Figure 4.2 for 5 different grid resolutions. The analytical solution by Prosperetti [52] is included as well. The amplitude error, i.e. the difference between the analytical and numerical solutions normalised by A_0 is shown in Figure 4.3. Table 4.1 summarises the root mean square of the amplitude error

$$\epsilon_{\text{RMS}} = \frac{1}{A_0} \sqrt{\frac{1}{n_{\text{end}}} \sum_{n=1}^{n_{\text{end}}} \left(A(t_n) - A_{\text{analytical}}(t_n)\right)^2} \tag{4.2}$$

and its convergence order. A convergence rate between first and second order is observed. For small grid resolutions the RMS errors are slightly higher than those reported by Desjardins et al. [18], but a better convergence is observed. Compared with the results by Herrmann [33] our results exhibit again a better convergence while the errors for a grid with 16 points in each direction are almost identical.

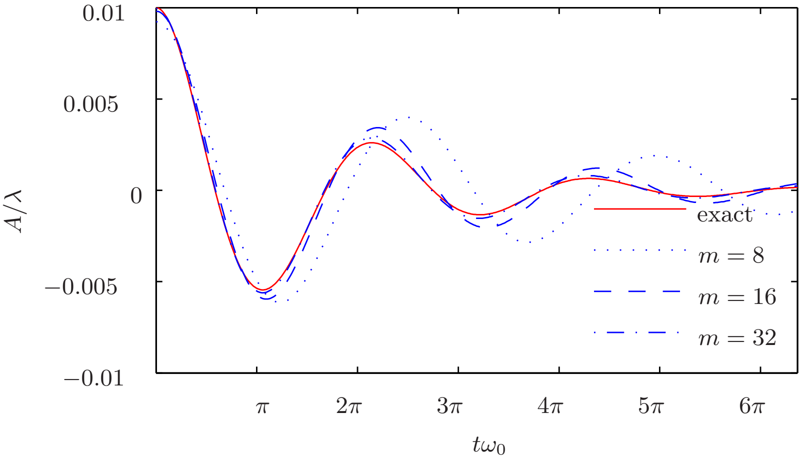

Figure 4.2: Time evolution of the amplitude of a damped interface wave.

The standard method presented in chapter 3 uses a nonconservative formulation of the advective terms in the Navier-Stokes equations (2.17). Alternatively one could rewrite the advective terms in conservative form, i.e. $\nabla \cdot (\mathbf{uu})$. This form must then be discretised by an appropriate method. Here we tested a 5th order finite difference WENO scheme using a Lax-Friedrichs flux splitting [61]. The conservative method is more complicated and requires more operations than the method presented in section 3.1.2. The reason is that for the nonconservative method the upwind direction is decided before the gradients are computed, while the conservative method has to compute the positive as well as the negative fluxes before the upwind direction can be identified. To assess the influence of the advective discretisation, the damped interface wave example was recomputed using the conservative discretisation. As it can be seen in Table 4.1, the results are almost identical to the standard discretisation.

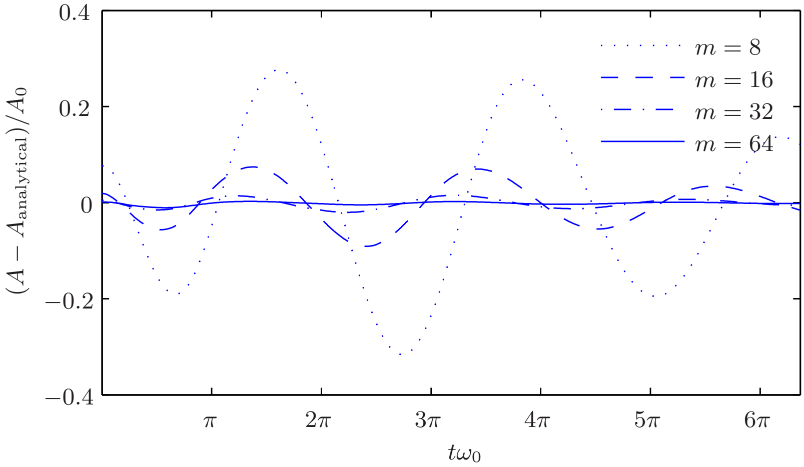

Figure 4.3: Amplitude error of a damped interface wave.

Another source for errors in the presented method is the computation of the jump conditions (2.33) and (2.38). To compute the jump conditions, the derivatives of the velocity need to be approximated. But those have a discontinuity in the first derivative. To omit the need to approximate the derivatives of the velocities with its discontinuities, one can move the normal or tangent inside the difference operator, see equation (2.39). Then the quantities on which the differential operators act are continuous. But also this approach will produce an error, since the discrete normals are stored only at the grid centres and their gradients are not zero. Again the numerical experiment for the standing wave was repeated, and the quantities in (2.33) and (2.38) where computed in the following way. First the velocity components where interpolated to the cell centres where they were multiplied by the appropriate normal and tangential components. The gradients of the resulting expressions were approximated using standard second order central finite difference stencils before the approximations were multiplied by the remaining normal and tangential components. The amplitude errors for the alternative jump computation are shown in Figure 4.4. The results are again similar to the standard method. From Table 4.1 it is visible that the errors are slightly higher, if the alternative method to compute the jumps is applied.

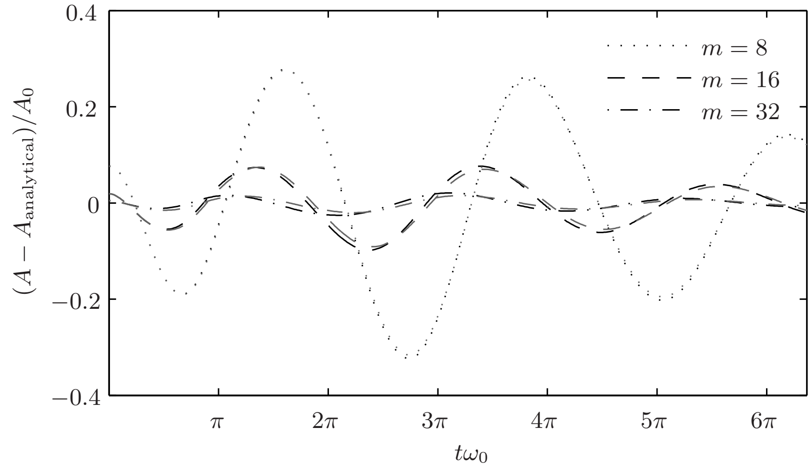

Figure 4.4: Amplitude error of a damped interface wave, where the normals are taken inside the differential operators. The grey lines indicate the errors of the standard method.

4.2 Contact line examples

The boundary condition for the momentum equation at the contact point should account for the slip between the wall and the contact point. Otherwise, the stresses would become singular around the contact point. To avoid this singularity, a number of authors have used a Navier condition (4.3) parallel to the wall with a constant slip length λ along the entire wall. An example of such a momentum boundary condition is given in section 4.2.2. The discrete no-slip boundary condition on a staggered grid corresponds to a Navier slip boundary condition with $\lambda = O(\Delta x^2)$ due to the interpolation error [2]. Dussan V [21] used an alternative approach, setting the velocity in the proximity of the contact point equal to the contact point velocity itself and letting it smoothly approach the wall velocity as the distance to the contact point increases. The length scale L_s of this smooth transition is of the order of the in-

Table 4.1: Amplitude errors ϵ_{RMS} and convergence rate $\mathcal{O}$ of the damped interface test for the standard method and two variants.

grid	standard		conservative		alternative jump condition	
	ϵ_{RMS}	$\mathcal{O}$	ϵ_{RMS}	$\mathcal{O}$	ϵ_{RMS}	$\mathcal{O}$
8×8	0.1693		0.1709		0.1727	
16×16	0.0454	1.87	0.0454	1.88	0.0482	1.79
32×32	0.0099	2.29	0.0099	2.30	0.0122	1.97
64×64	0.0034	1.46				
128×128	0.0025	0.67				

terface thickness. It was shown that the results for different velocity profiles only differ within a region of the extent of the order of L_s. Our preliminary numerical experiments revealed that a given wall velocity profile or a slip boundary condition with a non-zero λ close to the contact point will produce results which are similar away form the contact point, as long as the length scale L_s is kept constant. Both length scales λ and L_s are too small to be properly represented by the grids used in practical applications of the Navier-Stokes equations.

For the examples presented in this section a Navier boundary condition

$$\begin{aligned} \mathbf{u} \cdot \mathbf{t} &= \lambda \mathbf{t} \cdot \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) \cdot \mathbf{t} \\ \mathbf{u} \cdot \mathbf{n} &= 0, \end{aligned} \tag{4.3}$$

was used. Here $\mathbf{n}$ is a wall normal unit vector, $\mathbf{t}$ is a wall parallel unit vector and λ is the slip length, which is non-zero only in the vicinity of the contact point (except for section 4.2.2 where it is constant everywhere). Since almost complete slip is required around the contact point, the grid points directly to the right and left of the contact point have a large slip length which is reduced to zero over the next three grid points. The slip length is here set to $\lambda = 100$ for the two grid points adjacent to the contact point and is then reduced to $\lambda = 1$ and 0.01 over the two consecutive grid points on either side of the interface. For all the other grid points equation (4.3) with $\lambda = 0$ reduces to the ordinary no-slip boundary condition. As it is demonstrated by the results in this section, this boundary condition leads to converging results.

4.2.1 Capillary rise

We consider two immiscible fluids between two short parallel plates. The distance between the two plates is $2L$ their length is $3L$. At both ends of the channel ($x = 0$ and $x = 3L$) the pressure is fixed at $p = 0$. Gravity $g = 0.6$ is acting parallel to the wall in negative x-direction. The densities of the fluids are $\rho^+ = 1.0$ and $\rho^- = 0.1$. Both fluids have the same dynamic viscosity $\mu^+ = \mu^- = 0.01$. The surface tension between the fluids is $\sigma = 1.0$, while the static contact angle, measured in the fluid to the right of the interface, is $\theta_s = 160°$. The $+$ and $-$ are used to identify the left and right fluids, respectively. The contact angle θ_{CP} is always measured in the fluid to the right of the interface. All values are nondimensionalised using an appropriate combination of the basic units L for length, $\rho^+ L^3$ for mass and $\sqrt{\frac{\rho^+ L^3}{\sigma}}$ for time. A sketch of the considered test case is given in Figure 4.5.

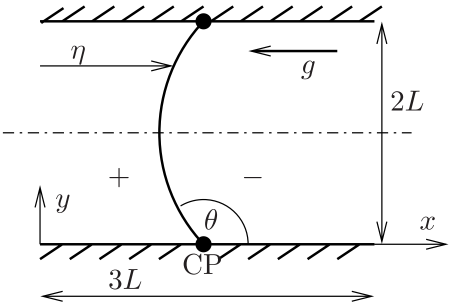

Figure 4.5: Sketch of the geometry of the capillary rise test.

The capillary rise is used to assess the accuracy of the contact point treatment. Since the stationary position of the contact point is sensitive to errors of the contact angle and curvature close to the wall, it is a good test for the method presented in section 3.4. Any mass flow over the interface, which occurs since the level set method is not mass conservative, is balanced by a mass flow at the top and bottom boundaries. Therefore the interface position is not affected by the mass loss, as it would be the case for a droplet which is completely immersed in another fluid.

First the analytical solution for the problem is presented. At steady state the capillary force will be balanced by the gravity force and the velocity will decay to zero, as the interface assumes a static position

$\eta(y)$. In both fluids we will have a hydrostatic pressure distribution:

$$p(x,y) = \begin{cases} p^+ = -x\rho^+ g & \text{if } x \leq \eta(y) \\ p^- = (h-x)\rho^- g & \text{if } x \geq \eta(y) \end{cases} , \tag{4.4}$$

where h is the extension of the capillary in the direction of the gravity. Thus the pressure jump at the interface is given as the difference of the pressure in the upper and lower fluid at $x = \eta(y)$:

$$\Delta p(y) = p^-(\eta(y), y) - p^+(\eta(y), y) = g\left(h\rho^- + \eta(y)\left(\rho^+ - \rho^-\right)\right) . \tag{4.5}$$

On the other hand the pressure jump at the interface is proportional to the interface curvature

$$\Delta p(y) = \sigma\kappa(y) = \sigma\frac{\eta''(y)}{\left(1+\eta'(y)^2\right)^{3/2}} . \tag{4.6}$$

Combining (4.5) and (4.6) we get the following ordinary differential equation for the interface position:

$$\eta''(y) = \frac{g}{\sigma}\left(1+\eta'(y)^2\right)^{3/2}\left(h\rho^- + \eta(y)\left(\rho^+ - \rho^-\right)\right) . \tag{4.7}$$

The boundary conditions are given by the static contact angle θ_s at $y = 0$ and the symmetry condition at $y = L$

$$\eta'(y=0) = -\tan(\theta_s - \pi/2) \tag{4.8}$$

$$\eta'(y=L) = 0 . \tag{4.9}$$

The ODE is solved numerically using the boundary value problem solver bvp4c in Matlab. At steady state the contact point is located at $x_{\mathrm{CP}} = 1.8789$. It is also pointed out that the stationary interface does not assume a circular shape like it is often assumed, if the height of a capillary rise is to be computed.

Since we are interested in the static position of the interface, an exact modelling of the dynamic behaviour around the contact point is not so important, which allows the use of a simple linear relation for the contact point velocity $f(\theta) = 1.1 \cdot (\theta_{\mathrm{CP}} - \theta_s)$. All simulations are initialised with zero velocity and a circular interface with a contact point position

$x_{\text{CP}} = 2.2$ and initial contact angle $\theta_{\text{CP}} = 120°$. The equations are integrated until $t = 50$. As opposed to the results in [f] the length of the domain is adjusted such that the distance between the first and last grid points (the centres of the first and last grid cells) corresponds to $3L$, because the Dirichlet boundary condition on the pressure is set at the cell centres. Since the problem is symmetric computations are only done for half of the channel, i.e. $0 \leq y \leq L$, with a symmetry boundary condition at $y = L$. For all results presented a uniform grid spacing of $h = 3L/m = L/n$ was used, where m and n are the number of grid points in x- and y-directions, respectively. The distance function was reinitialised 4 times per time unit. Each reinitialisation consisted of m reinitialisation steps with a CFL-number of 0.5. The interface position at $t = 50$ is shown in Figure 4.6 for four different grid sizes, while Figure 4.7 gives the difference between the numerical and exact solutions as a function of y. The interface error is computed as:

$$\epsilon_I = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(x_I(y_j) - \eta(y_j)\right)^2}\ , \tag{4.10}$$

where $x_I(y_j)$ is the interface position at y_j interpolated from the distance function. The contact point error $\epsilon_{\text{CP}} = |x_{\text{CP}} - \eta(0)|$ and the interface error ϵ_I are given in Table 4.2. They both exhibit a convergence order around 2, cf. the orders of convergence O_{CP} and O_I. Since the interface position is interpolated from the signed distance function ϕ the order of convergence for the interface error is one order higher than the overall order of convergence.

Table 4.2 also lists the maximum norm of the velocity components. The velocities at $t = 50$ are not zero as the analytical solution requires. The temporal evolution of the maximum norm of the x-component of the velocity is plotted in Figure 4.9. After the initial transient, where the contact point moves from $x = 2.2$ to its stationary position, the velocity starts to oscillate. From the right plot in Figure 4.9 it is evident that the frequency of those oscillations is the same as the reinitialisation frequency. Figure 4.8 shows the velocity field after a time step with reinitialisation. The interface is displaced slightly during each reinitialisation, causing a small change in the interface shape, which then will cause small changes in the interface curvature and angle close to the

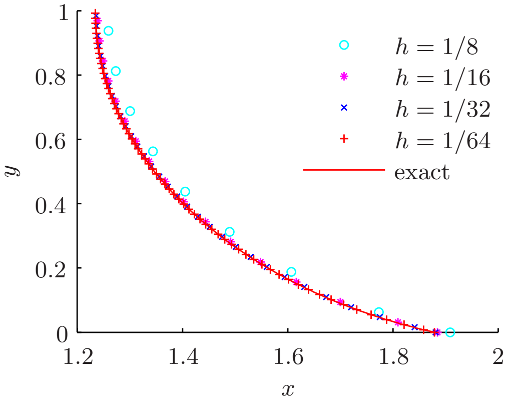

Figure 4.6: Interface position at steady state for capillary rise.

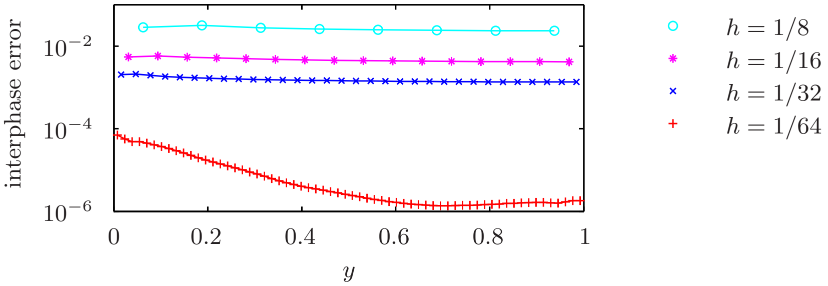

Figure 4.7: Interface error at steady state for capillary rise.

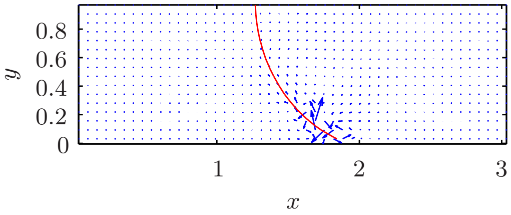

Figure 4.8: Velocity field and interface position for capillary rise at steady state immediately after reinitialisation for $h = 1/16$. The maximum norm of the velocity is given in Table 4.2.

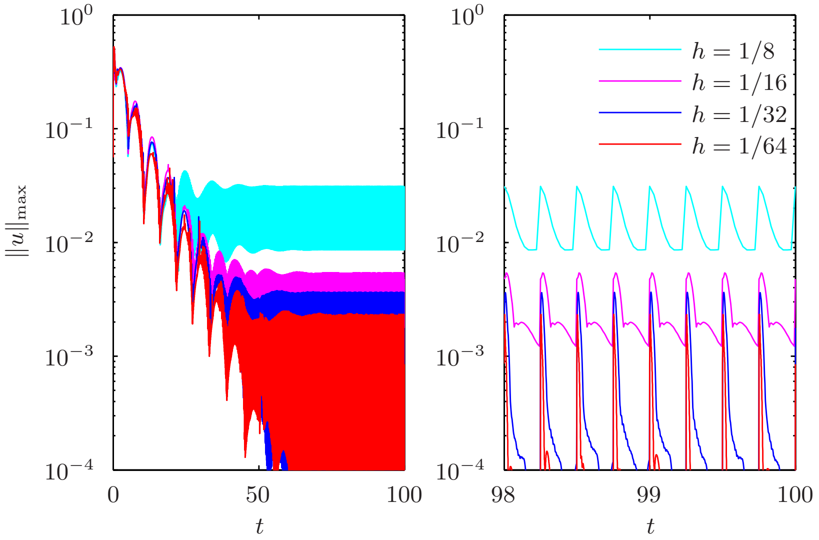

Figure 4.9: Maximum norm of the x-velocity component for various grid spacings h for capillary rise.

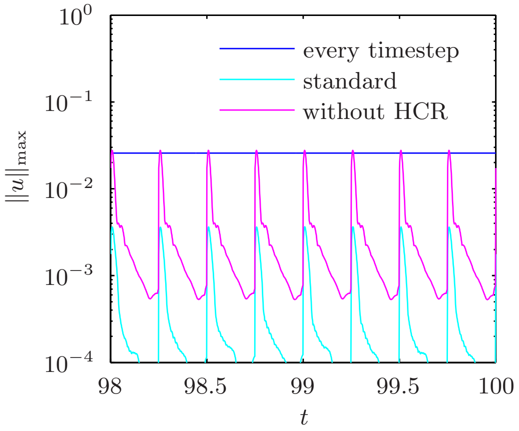

Figure 4.10: Maximum norm of the x-velocity component at steady state for different reinitialisation strategies for capillary rise. Grid spacing is $h = 1/32$.

Table 4.2: Measured errors for the capillary rise.

h	$\epsilon_{\text{CP}} \cdot 10^2$	O_{CP}	$\epsilon_I \cdot 10^2$	O_I	$\|u\|_{\max} \cdot 10^2$	$\|v\|_{\max} \cdot 10^2$
1/8	2.9962		2.6473		3.115	3.247
1/16	0.5763	2.4	0.4736	2.5	0.476	0.945
1/32	0.2108	1.5	0.1547	1.6	0.177	0.312
1/64	0.0052	5.3	0.0019	6.3	0.058	0.133

wall. As a result the velocity field in the region around the contact point is disturbed. In the time steps following the reinitialisation this disturbance is dissipated, i.e. the maximum norm of the velocity reduces rapidly until the next reinitialisation.

To demonstrate that it is important to use the constrained reinitialisation [32] in connection with the contact point treatment form section 3.4, the computations from above were repeated and the reinitialisation was done without HCR-1. The results are summarised in Table 4.3. For the coarse grids there is only a small difference in the interface and contact point position errors. As the grid is refined ϵ_{CP} and ϵ_I are not converging anymore. The velocity norm without HCR-1 increases about an order of magnitude. Table 4.3 includes also an example where 10 reinitialisation steps using HCR-1 were performed after each time step. The errors are higher than for the results where the reinitialisation is done with a time interval of 0.25. The temporal evolution of the velocity norm at steady state for the different reinitialisation strategies on a grid with $h = 1/16$ is plotted in Figure 4.10. The choice of the reinitialisation interval is a balance between introducing disturbances through the spurious displacement of the interface during the reinitialisation and the fact that the contact point and ϕ will lose their coupling without reinitialisation.

4.2.2 Gravity driven channel flow

A gravity driven channel flow is simulated to demonstrate the applicability of the presented method to a two-phase flow with a moving contact line. The test case also illustrates that the function $f(\theta_{\text{CP}})$ for the contact point velocity in (3.38) can easily be replaced by a relation,

Table 4.3: Measured errors for the capillary rise without constrained reinitialisation. The result marked with the star was computed with HCR, but every time step 10 reinitialisation steps were performed.

h	$\epsilon_{\mathrm{CP}} \cdot 10^2$	O_{CP}	$\epsilon_I \cdot 10^2$	O_I	$\|u\|_{\max} \cdot 10^2$	$\|v\|_{\max} \cdot 10^2$
1/8	3.2713		2.8978		6.469	5.445
1/16	0.5786	2.5	0.4450	2.7	2.700	3.707
1/32	0.5253	0.1	0.4032	0.1	1.730	2.421
1/32*	0.9589		0.6620		2.574	2.653

which is founded on physical principles. Ren and E [55] used molecular dynamics to derive a relation for the contact point velocity for a steady two-phase flow. They concluded that the appropriate boundary condition at the wall is given in (4.3), where $\lambda = \mu/\beta$ with β being the friction coefficient between the wall and a single fluid phase. According to their results the contact point velocity is given as

$$u_{\mathrm{CP}} = \frac{\sigma}{\beta_{\mathrm{CP}}} \left(\cos\left(\theta_s\right) - \cos\left(\theta_{\mathrm{CP}}\right)\right), \tag{4.11}$$

where β_{CP} is the effective contact point friction coefficient and θ_s denotes the static contact angle.

The same geometry and fluid properties as in Ren and E [55] were used, namely two immiscible fluids with identical properties confined by two parallel walls. The flow is driven by a gravity force g, which acts parallel to the wall in positive x-direction. The distance between the two plates is 200 and the length of the computation domain is chosen as 400. The fluid properties were measured in the molecular dynamics simulation by Ren and E [55]. The surface tension between the two fluids is $\sigma = 3.7$, the density and the dynamic viscosities of both fluids are $\rho = 0.81$ and $\mu = 2.2$, respectively. Since both fluids are the same, their interaction with the wall must be the same. Therefore the static contact angle is $\theta_s = 90°$. The friction factors were again measured from the molecular dynamics simulations and are $\beta = 6.0$ in both fluids away from the interface, while the effective contact point friction coefficient is $\beta_{\mathrm{CP}} = 30.24$. All values are expressed in terms of basic units of the Lennard-Jones potential used by Ren and E [55]. The simulations are initialised with a straight interface at $x = 200$ and a grid of 64×128 cells is used for all presented results. Since the flow is symmetric about $y = 100$, only the lower half of the channel is computed using a symmetry boundary condition at $y = 100$. To prevent the interface from leaving the computational domain, the contact point position is fixed. Instead the wall is moved with a velocity $u_{\text{wall}} = -u_{\text{CP}}$, where the contact point velocity u_{CP} is computed from equation (4.11).

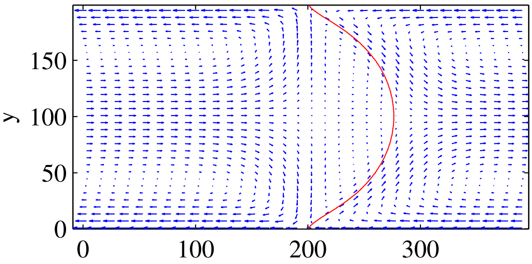

Figure 4.11: Gravity driven channel flow. Velocity field and interface position for $g = 4.4 \times 10^{-5}$. The velocities are given relative to the interface.

The steady state velocity field and the interface position are illustrated in Figure 4.11. The velocities are given in the frame of reference moving with the interface, i.e. the wall and the fluid close to the wall are moving in negative x-direction. It can be observed that there is a region with a large slip around the contact points. Away from the interface the flow is approaching an ordinary Poisseuile flow and the slip at the wall almost vanishes despite the fact that the friction factor β is constant along the entire wall. The large slip at the contact point is explained by the pressure jump at the interface, which leads to a large shear rate close to the contact point.

The contact angle θ_{CP} at steady state are 95.9°, 103.1°, 109.7° and 115.9° for $g = 1.1 \times 10^{-5}$, 2.2×10^{-5}, 3.3×10^{-5} and 4.4×10^{-5}, respectively. This corresponds to contact point velocities of $u_{\text{CP}} = 0.013$, 0.028, 0.041 and 0.054, respectively. The angles between the interface and the horizontal direction as a function of the y-position for various values of g are given in Figure 4.12. There is a clear difference between the static contact angle $\theta_s = 90°$ and the contact angle θ_{CP} in case of the moving contact line. Towards the centre of the channel the interface angle is first increasing rapidly until it reaches a maximum θ_m at the inflection point of the interface. After the inflection point the interface angle is decreasing almost linearly towards the centre of the channel, where it is 90° because of symmetry. The values of the contact angle θ_{CP} (and therefore also the contact point velocities) in our computations are slightly lower than reported by Ren and E [55]. On the other hand the maximum values of the interface angle are higher in the present results. The main difference between the two continuum solvers is the fact that Ren and E [55] smooth the surface tension force over multiple grid points, whereas in the present method the surface tension is implemented as a sharp jump in the normal stress. For the discussed flow configuration the surface tension plays a crucial role and a small deviation in the discretisation of the surface force could lead to the observed differences in the steady interface shape.

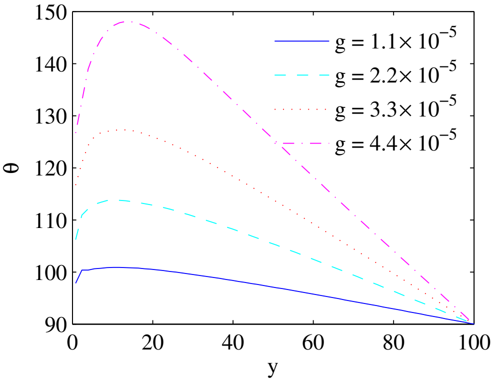

Figure 4.12: Gravity driven channel flow. Angle between the interface and the wall as a function of y for various values of g.

Finally the computed results are compared with the analytical results by Cox [15]. Cox provides an expression for the macroscopic contact angle, which can be associated with the maximum of the angle between the interface and the wall θ_m:

$$g(\theta_m) = g(\theta_{\mathrm{CP}}) + \mathrm{Ca}\ln(\epsilon^{-1}), \tag{4.12}$$

where $\mathrm{Ca} = \frac{\mu u_{\mathbf{CP}}}{\sigma}$ is the Capillary number and ϵ is the non dimensional slip length. The solid line in Figure 4.13 represents $g(\theta)$, whereas the symbols are $g(\theta_{\mathrm{CP}}) + a u_{\mathrm{CP}}$ for the four different simulation runs. A least squares fit gives the coefficient $a = 1.427$. Figure 4.13 demonstrates that, the numerical results fit well to the theoretical results by Cox [15] if $a = 1.427$.

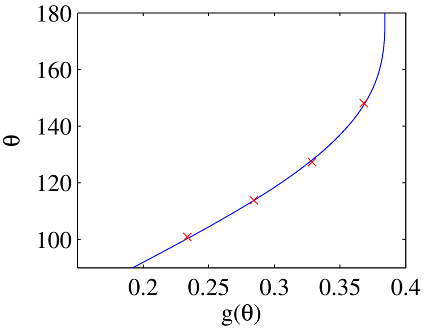

Figure 4.13: Gravity driven channel flow. The apparent contact angle fitted to Cox's theory. Solid line: $g(\theta)$, Symbols: θ_m against $g(\theta_{\text{CP}}) + 1.43u_{\text{CP}}$

4.2.3 Capillary driven channel flow

In the last two examples presented in the thesis, the relation f between the contact point velocity u_{CP} and contact angle θ_{CP} in (3.38) is the result of a micro scale simulation around the contact point. It is assumed that the time scale of the flow close to the contact point is much smaller than the time scale of the global problem. The scale separation both in time and space allows that the relation $f(\theta_{\text{CP}})$ in (3.38) for the contact point velocity can be precomputed for a number of contact angles. For the examples shown here the micro model was first evaluated for a number of contact angles and the resulting contact point velocity was tabulated [e]. Then a rational polynomial was fitted to the results from the micro model and this polynomial was then used as $f(\theta_{\text{CP}})$. In order to not infringe upon the assumption of temporal scale separation we used two examples where the driving force of the flow are the capillary effects at the contact line.

The first example is a capillary rise in a horizontal channel. In the following, we use non-dimensional variables. Two immiscible fluids are placed between two plates separated by a distance of 2. The length of the channel is 10 and the pressure is fixed to zero at the in- and outlet. There is no gravity acting on the fluids. Initially the interface between the two fluids is a straight line at $x = 3.5$. The grid spacing is h in both directions. After an initial transient, the interface assumes a curved shape and travels with an almost constant velocity towards the right as the viscous and capillary forces are in balance. The mass densities of the two fluids are $\rho^+ = 1$ and $\rho^- = 0.8$ and the dynamic viscosities are

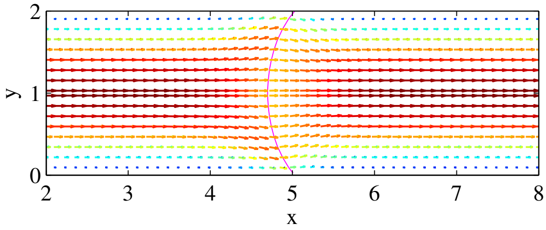

Figure 4.14: Capillary driven channel flow. Velocity field and interface position as the contact point passes $x = 5$. The maximum velocity corresponds to the interface velocity $\bar{v}_{\mathrm{CP}}$ given in Table 4.4.

$\mu^+ = 0.3$ and $\mu^- = 1$, respectively. The viscosity and density ratios correspond to the values for water and oil. The surface tension between the fluids is $\sigma = 1$ and the static contact angle, measured in the fluid with negative ϕ, is $\theta_s = 140°$. The fluid$^-$ is located to the right of the interface. The situation corresponds to two large vessels, each filled with a different fluid. The fluid level in both vessels is chosen such that the hydrostatic pressure at the bottom is the same in both vessels. At the bottom both vessels are connected by a small rectangular channel, which is initially blocked. As soon as the channel is opened, fluid$^+$ starts to displace fluid$^-$, i.e. the interface between the two fluids starts to move.

The velocity field and the interface position at a time after the initial transient are displayed in Fig. 4.14. In Fig. 4.15, the contact point velocity and position are plotted for a number of increasingly refined grids. The results are converging as the grid is refined. This is also evident from Table 4.4 where the time averaged contact point velocities obtained with increasing grid resolutions are summarised. In Fig. 4.15 a slow increase of the contact point velocity after the initial transient can be observed. This is due to the reduction of the overall viscous force as the less viscous fluid displaces the one with higher viscosity. The contact point velocity is oscillating considerably at lower resolution. As the resolution is increased, the amplitude of the oscillations is reduced while their frequency increases. The cause for the oscillations is the interpolation of the intersection point between the zero level set of ϕ and the grid lines which are used to compute the curvature at the contact point as well as the contact angle. The interpolation errors depend on

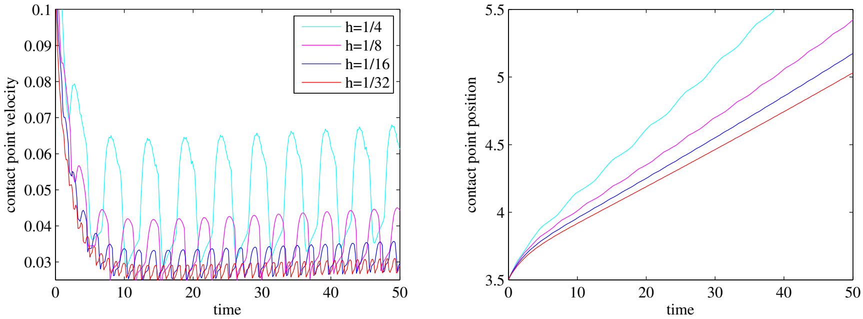

Figure 4.15: Contact point velocity (left) and position (right) for capillary driven channel flow with refined grids.

Table 4.4: Averaged velocity of the contact point $\bar{v}_{\mathrm{CP}}$ for different grid spacing h. The averaging is done over time t when $4.5 \leq x_{\mathrm{CP}}(t) \leq 5.5$. The result marked with the star was computed using a 10 times larger value of λ around the contact point.

h	1/4	1/8	1/16	1/32	1/16*
$\bar{v}_{\mathrm{CP}}$	0.0499	0.0362	0.0317	0.0291	0.0321

the distance between the intersection point and the next grid points. Consequently, the estimation of the contact angle changes as the interface is moving relative to the grid. With an increasing grid resolution, the errors in the contact angle approximation decrease and hence the oscillations in the contact point velocity are reduced.

In Table 4.4, we also included a result where λ in equation (4.3) has been multiplied by 10. The increase of λ causes a much smaller change in the result than doubling of the amount of grid points. Therefore it can be assumed that the described values of λ allow for enough slip around the contact point.

4.2.4 Advancing droplet

We consider the same fluids as in the previous example. Initially a droplet of the fluid$^+$ is placed on a flat surface. The initial interface shape is a semicircle with a radius of $R_0 = 1$ and a corresponding wall contact angle of $\theta_{\mathrm{CP}} = 90°$. The two contact points are located at $x_{\mathrm{CP}} = \pm 1$, cf. $t = 0$ in Figure 4.16 where only the right half of the interface is plotted. In order to minimise the surface energy, the drop will spread on the surface until the static contact angle is reached. In the final steady state, the interface will assume a circular shape with a radius of $R = \sqrt{\frac{\pi}{2(\pi-\theta_s)-\sin(2(\pi-\theta_s))}}$ and contact point positions $x_{\mathrm{CP}} = \pm R\sin(\pi - \theta_s)$. The velocities are zero at steady state. Because of symmetry, only half of the droplet is computed on a rectangular computational domain of $[0, 2.25] \times [0, 1.5]$, with the circle centre located at the origin. All computations were performed with equal grid spacing h in both dimensions. At $x = 0$, $x = 2.25$, and $y = 1.5$ symmetry boundary conditions are applied.

The evolution of the interface position as the droplet is spreading and the instantaneous velocity field at $t = 8$ are shown in Fig. 4.16. As the contact point advances towards the right, a vortex develops. The maximum absolute value of the velocity is always located around the contact point and the slip allowed by the boundary condition is clearly visible. The small capillary number $\mathrm{Ca} = \frac{\mu_2 U_{\mathrm{CP}}}{\sigma} \ll 1$ indicates that the capillary forces are dominating, and hence the interface shape resembles a circle. The deviation from a circular shape is more pronounced at initial times. This is because the large initial difference between the wall contact angle and the static contact angle leads to a higher contact point velocity U_{CP} and therefore larger capillary numbers. The temporal evolution of the contact point position is displayed in Fig. 4.17. As in the example above, grid refinement leads to a decrease of the contact point velocity. The mass loss of the level set method [60] is an important contribution to the error in the contact point position at steady state. Since the droplet loses mass over time, the contact point will reach a steady state position which is on the left of the analytically suggested value. It is also slowly receding further as the droplet continues to lose mass. This is clearly visible for the case with the coarse grid spacing $h = 1/8$.

On the basis of energy considerations, Ren et al. [56] derived a scaling law for the contact point position x_{CP} of a small advancing droplet. Their results show that $x_{\mathrm{CP}}(t) \sim t^{\alpha}$, where α is between 1/7 and 1/5 depending on whether the viscous forces in the fluids or the friction forces at one of the interfaces dominates the energy dissipation. The right plot in Fig. 4.17 shows the evolution of contact point position in a logarithmic scale. During the time interval $4 < t < 20$, the contact point position follows the scaling law by Ren et al. [56]. By a least square fit, α is identified as $1/7.0580$, $1/6.8015$, and $1/6.6959$ for grid spacing of $h = 1/8$, $h = 1/16$, and $h = 1/32$, respectively, which agrees well with the theoretical results.

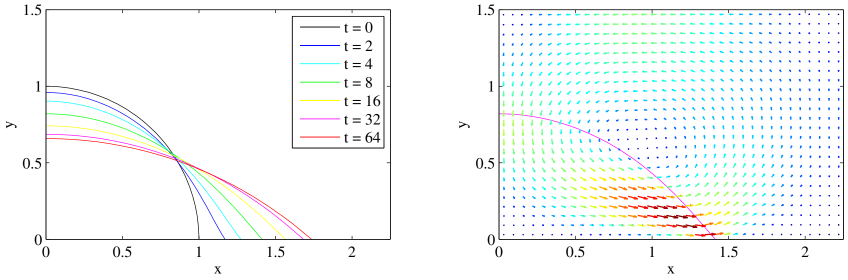

Figure 4.16: Interface position of the advancing droplet at different times t (left). Interface position and instantaneous velocity field at $t = 8$ (right).

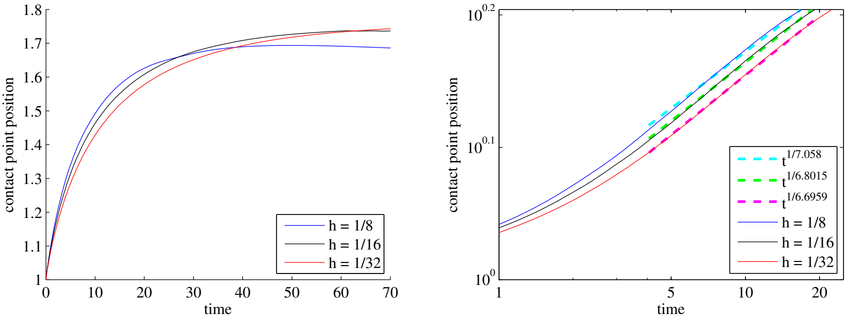

Figure 4.17: Contact point position of the advancing droplet in linear (left) and logarithmic scales (right).

Chapter 5

Conclusions and Outlook

5.1 Conclusions

A two-phase incompressible solver which treats the changes of the fluid properties and the surface tension at the interface in a sharp manner was implemented. The level set method was used to keep track of the interface position. The solver was tested and verified for two-phase-flow problems. Results for a damped surface wave are in good agreement with the results reported in the literature.

A method to treat contact points in the sharp interface solver was developed. A contact point is tracked explicitly along the wall, which lies half way between two grid points due to the staggered grid. The method allows a direct control of the motion of the contact points and is therefore very flexible in the sense that it can be used to implement different models to describe contact point dynamics. The important part in the procedure is to set appropriate boundary conditions for the signed distance function at the ghost points near the contact point. Moreover it was shown how to couple the contact point position to the Navier-Stokes equations through the curvature of the interface at the wall. This curvature at the wall has to be approximated carefully to get convergent results. A circle is fitted through the contact point and the intersection points of the zero contour line of the signed distance function with the first two grid lines parallel to the wall. The fitted circle is then used to calculate the contact angle and the curvature adjacent to the wall. The interpolation of the required intersection points has to be two orders more accurate than the desired curvature order. In addition care has to be taken that the interpolation stencil does not cross the discontinuity in the level set function, which is caused by the linearly prolonged interface. The present thesis gives examples such as a capillary rise and a gravity driven channel flow, which demonstrate the viability of the presented procedure to handle contact points.

The developed two-phase solver was used as a part of a multiscale framework for capillary driven multiphase flows. As a model for the micro scale behaviour around the contact point a phase field method with appropriate boundary conditions was used. The contact point in the macro solver was advected by the contact point velocity which was predicted by the phase field model for a given contact angle. Utilising the scale separation between the macro and micro models, it is possible for the micro model to use physical length scales over which contact line diffusion occurs. Thus the results from the phase field micro model represent real material behaviour. Therefore, our multiscale method circumvents the necessity to resolve the contact line processes at a detailed scale in the macro model by instead solving a micro scale model of moderate size. The macro scale flow solver can use considerably coarser meshes than a corresponding global phase field simulation would require, if it is to resolve all involved length scales. Thus the proposed multiscale method can give tremendous improvements in computational efficiency. Compared to other multiscale methods for contact line simulation using molecular dynamics, the phase field method is not restricted to fluids with equal properties. Numerical examples for an advancing drop and a capillary driven channel flow demonstrate the feasibility of the multiscale method.

In addition the thesis includes some work in relation to the conservative level set method (CLSM). A finite difference discretisation for the CLSM was implemented and examined. It was established that the interface is subject to a significant spurious deformation during reinitialisation, leading to a poor convergence. A stabilised constrained reinitialisation proved to improve the convergence of the reinitialisation, but it spoils the overall mass conservation property of the CLSM. A careful investigation of the curvature computation showed that despite the large gradient of the CLSM function the best curvature is obtained by direct differentiation of the CLSM function.

5.2 Outlook

Based on the work done for this thesis, the following suggestions for further work are given.

- An appropriate boundary condition for the momentum equation at the contact point should be developed using available information from the micro model. The goal is to avoid the arbitrary choice of the slip length λ at the contact point. As it is demonstrated by [55], a contact point increases the friction between the wall and the fluids. It should therefore be possible to measure a contact point related friction component in the phase field micro model, e.g. the integral of the additional wall parallel shear stress around the contact point. This integral value could then be used to set the boundary condition for the momentum equation in the macro model. One possible solution would be to choose the slip length at a few grid points surrounding the contact point such that the same integral friction force is produced as it was measured in the micro model.

- The extension of the multiscale methods for flows where the micro model is not in equilibrium should be considered. For these kinds of problems it will probably not be possible anymore to precompute the reaction of the micro model. Therefore both models have to be integrated in time together. It should also be investigated whether additional exchange of information between the two scales is necessary in addition to the contact point velocity and angle, which are exchanged in the presented multiscale method.

- The proposed method to control the motion of the contact point with an arbitrary function and the associated procedure to compute the values of the signed distance function at the ghost points could be extended to solid surfaces which are not aligned with the Cartesian grid using the method proposed by Liu et al. [46].

- For the two-phase solver it would be desirable to increase the convergence order at the jump. To achieve this, a better method to decompose the known interface normal jump conditions into the grid parallel components is required.

- An extension of the proposed contact point treatment to three dimensions is not straightforward, but in principle it is not impossible. Instead of keeping track of the position of a single point, in three dimensions a line has to be tracked. This could be done using similar algorithms, which are employed in front-tracking methods.
- The spurious displacement of the interface during the reinitialisation of the CLSM using a finite difference implementation remains an open issue.

Chapter 6

Summary of thesis articles

Article [a]

C. Walker and B. Müller. A conservative level set method for sharp interface multiphase flow simulation. In *Proceedings of ECCOMAS CFD 2010 Conference, Lisbon*, 2010.

A finite difference method for the computation of the interface location in two two phase flows is presented. The level set method is often used to describe the interface position. It's major drawback is that it does not conserve the mass of the fluids. To address this problem we use a finite difference implementation of the conservative level set method. The conservative level set method allows to write the advection and the reinitialisation as conservation laws. High order methods are employed for the advection of the level set function, to ensure an accurate representation of the interface location and to keep the shape of the level set function close to its hyperbolic tangent shape, which minimises the effort for the reinitialisation. Also the reinitialisation is discretised with high order finite difference methods.

As a result an explicit method which is relatively easy to implement and has favourable properties for two phase flows is obtained. It employees the same discretisation schemes which are used for the advection of an ordinary level set method. In addition the extension to three dimensions is straightforward. We present results computed with our method and compare them with the results from the original methods. Especially the improvements in mass conservation will be discussed.

Article [b]

C. Walker. Curvature computation for a sharp interface method using the conservative level set method. In A. Eriksson and G. Tibert, editors, *Proceedings of NSCM-23*, pages 203–206, 2010.

Different approaches for the computation of the curvature from the conservative level set function at the interface are compared. The first method computes the curvature at the grid points by approximating the derivatives of the conservative level set function. Then the curvature is interpolated to the interface position. The second method uses a distance function which is computed from the conservative level set function by a fast marching method to approximate the derivatives at the grid points. The third method computes the curvature directly at the interface position by fitting a local polynomial through neighbouring grid intersections. The accuracy of the resulting curvature is evaluated at the locations where they are used in the ghost fluid method for two-phase flows. In our test the method which computes the curvature from approximated derivatives and then interpolates it to the intersections is the most accurate.

Article [c]

C. Walker and B. Müller. Contact line treatment with the sharp interface method. In B. Skallerud and H. Andersson, editors, *MekIT'11: Sixth national conference on Computational Mechanics*, pages 451–462. Tapir Academic Press Trondheim, 2011.

The paper presents a method to handle contact points in a sharp interface method. The contact points are tracked explicitly and their velocities are a function of the contact angle. For sharp interface methods an accurate approximation of the curvature is important. This cannot be obtained by using the conventional central difference methods adjacent to walls since the level set functions are not defined therein. Therefore it is proposed to use the tracked contact point and the intersection points of the first two grid lines parallel to the wall with the zero contour line of the signed distance function to approximate the contact angle and curvature adjacent to the walls.

Article [d]

C. Walker and B. Müller. Constrained reinitialisation of the conservative level set method. In J. Olsen and S. Johansen, editors, *Proceedings of the 8th International Conference on CFD in Oil & Gas, Metallurgical and Process Industries*, 2011.

In this paper it is shown that the reinitialisation of the conservative level method (CLSM) introduces spurious displacements of the interface. The high order constrained reinitialisation method (HCR) prevents the interface in conventional level set methods from moving during the reinitialisation. Since the standard HCR is unstable for level set functions with large absolute values of the gradient, it has to be adopted for the CLSM. It is shown that HCR can be stabilised with an adaptive distribution of the forcing and that it retains the shape of the interface during reinitialisation of the CLSM. However, our numerical experiments show that fixing of the interface during reinitialisation spoils the mass conservation of the conservative level set method.

Article [e]

M. Kronbichler, C. Walker, G. Kreiss, and B. Müller. Microscale enhancement of macroscale modeling for capillary-driven contact line dynamics. Submitted, 2012.

A multiscale method for the simulation of the two-phase of immiscible incompressible fluids with moving contact lines is presented. The presented multiscale method is restricted to flows driven by capillary forces, with a clear separation between the scales of contact behaviour and global fluid flow. The system is represented by a macro model using the methods presented in [c] and [f], and the contact point velocity depends on the wall contact angle. The relation between this slip velocity and the wall contact angle is found by a micro model, based on the response of the flow to the molecular forces induced by the macroscopic contact angle. The dimensions of the micro simulation correspond to physical length scales over which contact line diffusion occurs, which makes the results from the phase field micro model represent real material behaviour. Numerical examples that demonstrate the viability of the approach are presented. A comparison with phase field results shows that the multiscale method is able to represent both transient and steady state behaviour in a capillary rise. Agreement of the results with previous studies of advancing droplets is demonstrated as well.

Article [f]

C. Walker and B. Müller. Contact line treatment with the sharp interface method. Submitted, 2012.

This paper presents an improvement of the technique to handle contact points in a sharp interface method from [c]. For sharp interface methods an accurate approximation of the curvature is important. However, this cannot be obtained by using the conventional central difference methods adjacent to walls since the level set functions are not defined therein. Therefore we propose to use the tracked contact point and the intersection points of the first two grid lines parallel to the wall with the zero contour line of the signed distance function to approximate the contact angle and curvature adjacent to the walls. It is shown that interpolation of the intersection points for the curvature calculation, must omit the discontinuity in the signed distance function. The coupling between the contact points and the zero level set is enforced by boundary conditions during reinitialisation of the distance function. The signed linear extrapolation is introduced to compute the ghost point values, where the characteristics do not demand a boundary condition. Applications to capillary rise and a gravity driven two phase channel flow show the validity of the approach and its grid convergence.

Bibliography

[1] A. W. Adamson and A. P. Gast. *Physical chemistry of surfaces.* Wiley New York, 6th edition, 1997.

[2] S. Afkhami, S. Zaleski, and M. Bussmann. A mesh-dependent model for applying dynamic contact angles to vof simulations. *Journal of Computational Physics*, 228(15):5370–5389, 2009.

[3] D. M. Anderson, G. B. McFadden, and A. A. Wheeler. Diffuse-interface methods in fluid mechanics. *Annual Review of Fluid Mechanics*, 30(1):139–165, 1998.

[4] T. A. Baer, R. A. Cairncross, P. R. Schunk, R. R. Rao, and P. A. Sackinger. A finite element method for free surface flows of incompressible fluids in three dimensions. part II. Dynamic wetting lines. *International Journal for Numerical Methods in Fluids*, 33 (3):405–427, 2000.

[5] T. D. Blake. The physics of moving wetting lines. *Journal of Colloid and Interface Science*, 299(1):1 – 13, 2006.

[6] D. Bonn, J. Eggers, J. Indekeu, J. Meunier, and E. Rolley. Wetting and spreading. *Reviews of Modern Physics*, 81(2):739–805, 2009.

[7] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335 – 354, 1992.

[8] M. Bussmann. *A three-dimensional model of an impacting droplet.* PhD thesis, University of Toronto, 2000.

[9] J. W. Cahn and J. E. Hilliard. Free energy of a nonuniform system. I. Interfacial free energy. *Journal of Chemical Physics*, 28(2):258–267, 1957.

[10] D. Caviezel, C. Narayanan, and D. Lakehal. Adherence and bouncing of liquid droplets impacting on dry surfaces. *Microfluidics and Nanofluidics*, 5:469 – 478, 2008.

[11] Y. A. Çengel and J. M. Cimbala. *Fluid Mechanics Fundamentals and Applications*. McGraw-Hill, Boston, 2nd edition, 2010.

[12] 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(2):449 – 464, 1996.

[13] A. J. Chorin and J. E. Marsden. *A Mathematical Intro to Fluid Mechanics*, volume 4 of *Texts in Applied Mathematics*. Springer Verlag, New York, 1993.

[14] K. T. Chu and M. Prodanovic. Lsmlib. Website, 2009. URL `http://ktchu.serendipityresearch.org/software/lsmlib/index.html`.

[15] R. G. Cox. The dynamics of the spreading of liquids on a solid surface. Part 1. Viscous flow. *Journal of Fluid Mechanics*, 168: 169–194, 1986.

[16] P. G. De Gennes. Wetting: statics and dynamics. *Reviews of Modern Physics*, 57(3):827, July 1985.

[17] D. Deganello, T. N. Croft, A. J. Williams, A. S. Lubansky, D. T. Gethin, and T. C. Claypole. Numerical simulation of dynamic contact angle using a force based formulation. *Journal of Non-Newtonian Fluid Mechanics*, 166(16):900 – 907, 2011.

[18] O. Desjardins, V. Moureau, and H. Pitsch. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of Computational Physics*, 227(18):8395–8416, 2008.

[19] H. Ding and P. D. M. Spelt. Inertial effects in droplet spreading: a comparison between diffuse-interface and level-set simulations. *Journal of Fluid Mechanics*, 576:287–296, 2007.

[20] D. Drikakis and W. Rider. *High-Resolution Methods for Incompressible and Low-Speed Flows.* Springer-Verlag Berlin Heidelberg, 2005.

[21] E. B. Dussan V. The moving contact line: the slip boundary condition. *Journal of Fluid Mechanics*, 77(04):665–684, 1976.

[22] W. E, B. Engquist, and Z. Huang. Heterogeneous multiscale method: A general methodology for multiscale modeling. *Phys. Rev. B*, 67(9):092101, March 2003.

[23] W. E, B. Engquist, X. Li, W. Ren, and E. Vanden-Eijnden. The heterogeneous multiscale method: A review. *Communications in Computational Physics*, 2(3):367–450, June 2007.

[24] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1):83–116, 2002.

[25] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152 (2):457–492, 1999.

[26] T. P. Fries. The intrinsic XFEM for two-fluid flows. *International Journal for Numerical Methods in Fluids*, 60(4):437–471, 2009.

[27] P. R. Gunjal, V. V. Ranade, and R. V. Chaudhari. Dynamics of drop impact on solid surface: Experiments and VOF simulations. *AIChE Journal*, 51(1):59–78, 2005.

[28] N. G. Hadjiconstantinou. *Hybrid atomistic-continuum formulations and the moving contact line problem.* PhD thesis, Massachusetts Institute of Technology. Dept. of Mechanical Engineering, 1998.

[29] E. B. Hansen. *Numerical Simulation of Droplet Dynamics in the Presence of an Electric Field.* PhD thesis, Norwegian University of Science and Technology, 2005.

[30] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8(8):2182, 1965.

[31] D. Hartmann, M. Meinke, and W. Schröder. Differential equation based constrained reinitialization for level set methods. *Journal of Computational Physics*, 227(14):6821 – 6845, 2008.

[32] D. Hartmann, M. Meinke, and W. Schröder. The constrained reinitialization equation for level set methods. *Journal of Computational Physics*, 229(5):1514 – 1535, 2010.

[33] M. Herrmann. A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *Journal of Computational Physics*, 227(4):2674 – 2706, 2008.

[34] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225, January 1981.

[35] C. Huh and L. Scriven. Hydrodynamic model of steady movement of a solid/liquid/fluid contact line. *Journal of Colloid and Interface Science*, 35(1):85 – 101, 1971.

[36] J. N. Israelachvili. *Intermolecular and surface forces*. Elsevier Amsterdam, 3rd edition, 2011.

[37] D. Jacqmin. Contact-line dynamics of a diffuse fluid interface. *Journal of Fluid Mechanics*, 402:57–88, 2000.

[38] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202 – 228, 1996.

[39] S. T. Johansen, T. Ytrehus, and K. E. Einarsrud. Modeling of multiphase flows. Lecture notes, Norwegian University of Science and Technology, 2012.

[40] M. Kang, R. P. Fedkiw, and X.-D. Liu. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, 15(3):323–360, 2000.

[41] J. Kim. Phase-field models for multi-component fluid flows. *Communications in Computational Physics*, 12(3):613–661, 2012.

[42] J. Koplik, J. R. Banavar, and J. F. Willemsen. Molecular dynamics of fluid flow at solid surfaces. *Physics of Fluids A: Fluid Dynamics*, 1(5):781–794, 1989.

[43] P. K. Kundu and I. M. Cohen. *Fluid Mechanics.* Elsevier Academic Press, San Diego, 3rd edition, 2004.

[44] L. Leger and J. F. Joanny. Liquid spreading. *Reports on Progress in Physics*, 55(4):431, 1992.

[45] Z. Li and M.-C. Lai. The immersed interface method for the Navier-Stokes equations with singular forces. *Journal of Computational Physics*, 171(2):822 – 842, 2001.

[46] H. Liu, S. Krishnan, S. Marella, and H. Udaykumar. Sharp interface Cartesian grid method II: A technique for simulating droplet interactions with surfaces of arbitrary shape. *Journal of Computational Physics*, 210(1):32–54, November 2005.

[47] 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(1):151–178, May 2000.

[48] B. J. Miller. *Improvements in Computing Multiple Phase Flows.* PhD thesis, University of California, Los Angeles, 1997.

[49] E. Olsson and G. Kreiss. A conservative level set method for two phase flow. *Journal of Computational Physics*, 210(1):225–246, 2005.

[50] E. Olsson, G. Kreiss, and S. Zahedi. A conservative level set method for two phase flow II. *Journal of Computational Physics*, 225(1): 785–807, 2007.

[51] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied Mathematical Sciences.* Springer-Verlag, New York, 2003.

[52] A. Prosperetti. Motion of two superposed viscous fluids. *Physics of Fluids*, 24(7), 1981.

[53] T. Qian, X.-P. Wang, and P. Sheng. Molecular scale contact line hydrodynamics of immiscible flows. *Phys. Rev. E*, 68(1):016306, 2003.

[54] W. Ren and W. E. Heterogeneous multiscale method for the modeling of complex fluids and micro-fluidics. *Journal of Computational Physics*, 204(1):1–26, March 2005.

[55] W. Ren and W. E. Boundary conditions for the moving contact line problem. *Physics of Fluids*, 19:022101, February 2007.

[56] W. Ren, D. Hu, and W. E. Continuum models for the contact line problem. *Physics of Fluids*, 22(10):102103, 2010.

[57] 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(1):243–263, July 2001.

[58] G. Russo and P. Smereka. A remark on computing distance functions. *Journal of Computational Physics*, 163(1):51 – 67, 2000.

[59] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31 (1):567–603, 1999.

[60] J. A. Sethian and P. Smereka. Level set methods for fluid interfaces. *Annual Review of Fluid Mechanics*, 35(1):341–372, 2003.

[61] C.-W. Shu. In A. Quarteroni, editor, *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, chapter Essentially Non-oscillatory and Weighted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws, pages 325–432. Springer, Berlin / Heidelberg, 1998.

[62] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439 – 471, 1988.

[63] P. D. Spelt. A level-set approach for simulations of flows with multiple moving contact lines with hysteresis. *Journal of Computational Physics*, 207(2):389 – 404, 2005.

[64] 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(2):301–337, August 2000.

[65] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114(1):146–159, September 1994.

[66] P. A. Thompson. *Compressible-fluid dynamics.* McGraw-Hill book company, 1972.

[67] 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 computations of multiphase flow. *Journal of Computational Physics*, 169(2):708–759, May 2001.

[68] G. Tryggvason, R. Scrdovelli, and S. Zaleski. *Direct Numerical Simulations of Gas-Liquid Multiphase Flows.* Cambridge University Press, Cambridge, 2011.

[69] W. Villanueva and G. Amberg. Some generic capillary-driven flows. *International Journal of Multiphase Flow*, 32(9):1072 – 1086, 2006.

[70] v. Šikalo, H.-D. Wilhelm, I. V. Roisman, S. Jakirlić, and C. Tropea. Dynamic contact angle of spreading droplets: Experiments and simulations. *Physics of Fluids*, 17(6), 2005.

[71] T. Young. An essay on the cohesion of fluids. *Philosophical Transactions of the Royal Society of London*, 95:65 – 87, 1805.

[72] P. Yue, C. Zhou, and J. J. Feng. Sharp-interface limit of the cahn-hilliard model for moving contact lines. *Journal of Fluid Mechanics*, 645:279–294, 2010.

Reasearch articles in full text

Article [a]

A conservative level set method for sharp interface multiphase flow simulation

C. Walker and B. Müller

In *Proceedings of ECCOMAS CFD 2010 Conference, Lisbon*, 2010

V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

A CONSERVATIVE LEVEL SET METHOD FOR SHARP INTERFACE MULTIPHASE FLOW SIMULATION

Claudio Walker*, **Bernhard Müller***

*Norwegian University of Science and Technology
Department of Energy and Process Engineering
N-7491 Trondheim, Norway
e-mail: claudio.walker@ntnu.no
e-mail: bernhard.muller@ntnu.no

Key words: Conservative level set, WENO, Interface capturing

Abstract. *We present a finite difference method for the computation of the interface location in two two phase flows. The method is especially suited for flow situations where the surface tension plays an important role. Examples for such flows are falling droplets, liquid jets or applications in micro fluidics..*

The level set method is often used to describe the interface position. It's major drawback is that it does not conserve the mass of the fluids. To address this problem we use a finite difference implementation of the conservative level set method. The signed distance function is replaced by a hyperbolic tangent function. This allows to write the advection and the reinitialisation as conservation laws. High order methods are employed for the advection of the level set function, to ensure an accurate representation of the interface location and to keep the shape of the level set function close to to its hyperbolic tangent shape, which minimises the effort for the reinitialisation.

As a result we get an explicit method which is relatively easy to implement and has favorable properties for two phase flows. As it emploies the same discretisation schemes which are used for the advection of an ordinary level set method it is relatively easy to implement in an existing two phase solver. In addition the extension to three dimensions is straightforward. We present results computed with our method and compare them with the results from the original methods. Especially the improvements in mass conservation will be discussed.

1 Introduction

Drop impacts on dry surfaces can be found in many industrial and natural processes. Applications where the behavior of the impacting drops plays an important role include ink-jet printing, spray cooling, pesticide spraying, erosion processes due to rain and thermal spray coating. Drop impacts also play an important role in gas-liquid separation in the process industry and the gas and oil industry. In these processes the surface tension plays often an important role. The spreading of the drop on the target surface is driven by inertia forces, whereas the surface tension acts against the spreading. In certain cases the surface tension is strong enough such that the drop can bounce back from the target. In order to simulate such flows it is important to have an accurate description of the surface tension force.

The surface tension in multiphase flows introduces a jump in the normal stress across the interface. A widely used method to deal with such flows is the continuous surface force (CSF) model. The surface tension force and the change of the fluid properties e.g. density and viscosity are smeared out over several grid points. The result of those diffuse interface methods are numerically smooth solutions, which in turn make it possible to apply standard finite difference methods. The ghost-fluid method (GFM) [1], which uses a fixed Cartesian grid, was extended to incompressible two-phase flows [2]. The GFM allows to retain the jumps across the interface and therefore it is possible to eliminate the error which stems from the artificial smearing of the fluid properties. To use the GFM it is necessary to know the location of the interface, as well as its curvature, a level set approach [3] is working well to retrieve this geometrical information about the interface. Another advantage of the level set is its ability to handle topological changes of the interface. However, the level set method has an important disadvantage, it does not conserve the mass of the two fluids [4]. Different approaches were developed to satisfy the mass conservation of the level set method. Examples include the conservative level set method [5] [6], the particle level set method (PLS) [7] or the coupled level set/volume-of-fluid (CLSVOF) [8]. The added complexity for both PLS and CLSVOF are significant. On the other hand the conservative level set methods improves the mass conservation and keeps the simplicity of the original method.

The main idea of the conservative level set method is to replace the signed distance function from the traditional level set method with a hyperbolic tangent profile. As a result the conservative level set method can be advected and reinitialized by conservative numerical methods.

1.1 Conservative level set method

In level set methods the interface is defined as the iso contour of a smooth function. For ordinary level set methods this function is the signed distance form the interface, and the interface location is where the distance function is zero. The conservative level set function replaces the distance function by a hyperbolic tangent function ϕ with values between zero and one. The position of the interface is located at the $\phi = 0.5$ contour line. Since we have smooth functions which are defined in the entire computational domain in both cases, we can easily extract additional geometrical informations about the interface. For example the interface normals $\mathbf{n}$ and the curvature κ are defined as

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \tag{1}$$

and

$$\kappa = \nabla \cdot \mathbf{n}. \tag{2}$$

The interface is transported simply by advecting the level set function $\phi_t = -\mathbf{u} \cdot (\nabla\phi)$. If we have a divergence free velocity field, as it is the case for incompressible flow, the interface transport can be written as a conservation law.

$$\frac{\partial\phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0 \tag{3}$$

Since all numerical methods will introduce an error as ϕ is advected, it will loose its hyperbolic tangent shape. The diffusion of the advection schemes will increase the distance in which ϕ rises from zero to one. Ollson and Kreiss [6] propose the following reinitialisation equation to force ϕ back to its hyperbolic tangent shape:

$$\frac{\partial\phi}{\partial\tau} + \nabla \cdot (\phi(1-\phi)\hat{\mathbf{n}} - \epsilon\left((\nabla\phi \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}\right)) = 0, \tag{4}$$

where $\hat{\mathbf{n}}$ are the normals at the beginning of the reinitialisation, and ϵ determines the width of the hyperbolic tangent. It is important to note that also the reinitialisation equation is a conservation law. The first flux term causes a compression of the profile, whereas the second term is a diffusive flux. By multiplications with the normals $\hat{\mathbf{n}}$ there are only fluxes in the direction of the normals. This forced flux direction for both the compression and the diffusion term are essential to improve the mass conservation of the method. To illustrate the nature of Equation (4), we use a 1 dimensional example. Suppose that the interface is located at $x = 0$, then the normals reduce to $n = -1$ or $n = 1$, in the example we use the latter. In this case a steady state solution to Equation (4) is:

$$\phi = \frac{1}{2}\left(1 + \tanh\left(\frac{x}{2\epsilon}\right)\right) \tag{5}$$

The solution is shown Figure 1, together with the compression and the diffusion term. It is clearly visible that at steady state the compression and the diffusion are balanced. If the ϕ would be too diffusive the compression term would outweigh the diffusive term and ϕ would be forced back to the steady state solution.

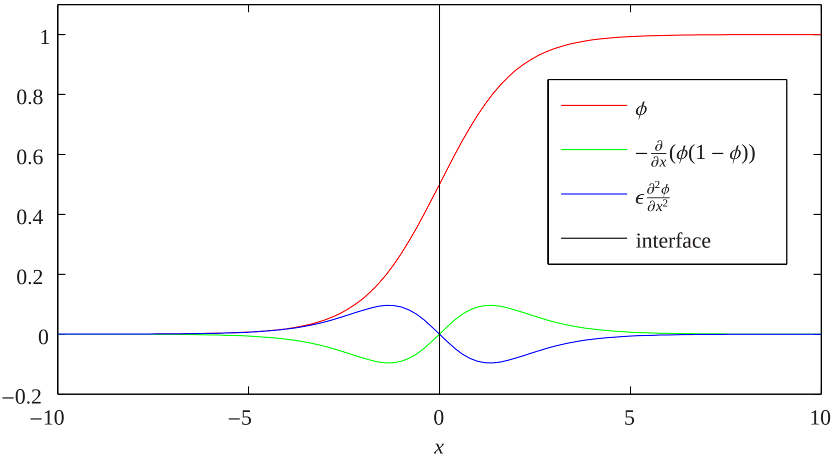

Figure 1: Illustration of the reinitialisation equation

2 Numerical Method

2.1 Advection

Since the conservative level set method will be a part of a multiphase solver, which uses the GFM to describe the jumps at the interface in a sharp way, it is a natural choice to use finite differences for its discretization. The spacial discretization for the advection Equation (3) is done with a standard 5th order finite difference WENO method as described in [9]. The advantage of the WENO scheme is that they do not produce artifical oscillations and therefore keep ϕ between zero and one. At the boundaries zero flux is imposed.

2.2 Normals

Before the reinitialisation Equation (4) can be solved the normals need to be computed. Far away from the interface the gradient of ϕ will be very small. As a result of the small gradients the direction of the normals will be extremely sensitive to small spurious errors in ϕ. If Equation (1) would be discretised directly using central differences with one sided stencils at the boundary, the resulting normals would point in arbitrary directions. Especially near the boundaries of the computation domain this problem will be amplified since the one sided stencils are less accurate. Such arbitrary normals pointing towards each other will lead to the accumulation of ϕ at wrong places during the reinitialisation.

Desjardins et. al. [10] propose to compute a singed distance function from ϕ using the fast marching method (FMM)[11]. The FMM is an efficient method to reinitialize the signed distance function ϕ_d for ordinary level set methods. The signed distance function has a gradient with unity length everywhere in the computation domain. Therfore the computation of the normals using (1) where ϕ is replaced by ϕ_d will be much more robust with regard to small errors in ϕ_d. We use a the FMM from the LSMLIB [12] which is second order in the L_2-norm to compute ϕ_d from ϕ. The normals $\hat{\mathbf{n}}$ are then computed from ϕ_d using a 4th order summation by parts (SBP) operator [13], with one sided stencils at the boundary.

2.3 Reinitialisation

Using twice a central difference approximation for the first derivative will not damp oscillations with a wavelength of $2\Delta x$. But the diffusive term in Equation (4) can not be computed using a central stencil for the second derivative because of the multiplications with the normals. Therefore $\nabla\phi$ is computed with a 4th order SBP operator and then the divergence total flux $\phi(1-\phi)\hat{\mathbf{n}} - \epsilon\left((\nabla\phi \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}\right)$ is approximated by the same 5th order WENO scheme which is used for the advection. Again a zero flux boundary condition is enforced.

3 Examples

To calculate the area inside the level set contour an unbiased level set contouring is used as it is described in [14]. This method is only second order accurate. The error of the interface location is measured with

$$\frac{1}{L}\int |\mathrm{H}(\phi_{expected}) - \mathrm{H}(\phi_{computed})| \,\mathrm{d}A. \tag{6}$$

Where L is the length of the interface and $\mathrm{H}(\phi) = 0$ for $\phi \leq 0.5$ or $\mathrm{H}(\phi) = 1$ otherwise. The numerical calculation of the integral is done as described in [3].

In all examples the forward Euler scheme is used for the time discretisation. Every 1000 time steps we perform 20 reinitialisation steps.

3.1 Vortex test

A stream function of

$$\psi(x, y, t) = \frac{1}{\pi}\sin^2(\pi x)\cos^2(\pi y)\cos(\pi t/T) \tag{7}$$

is given in a square unit domain. Initially a circle with a diameter of 0.3 is placed at $(0.5, 0.75)$. The circle will be transported in the vortex and reach its maximum deformation at $t = T/2$. From then on the velocity components will change their sign and the vortex should reach its initial position at $t = T$. The time step Δt is set to $5 \cdot 10^{-5}$ and the width of the hyperbolic tangent is $\epsilon = 0.8\Delta x$. In the literature two common values for T can be found, $T = 2$ will not lead to very thin filaments and is therefore often used for to show the method's ability at low numbers of grid points. On the other hand $T = 8$

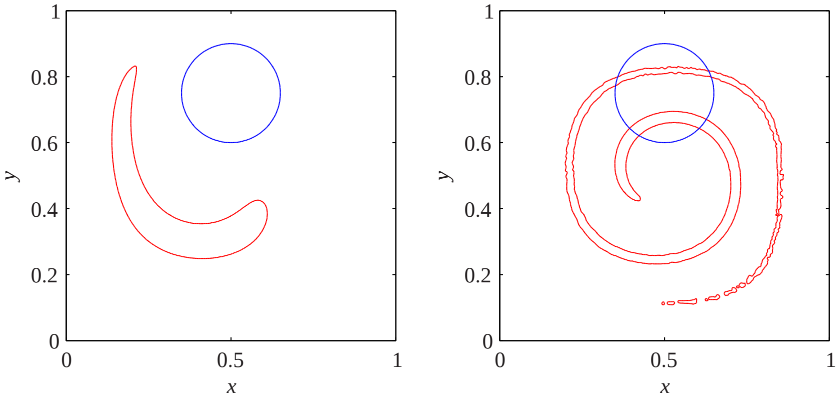

Figure 2: Circle undergoing deformation in a vortex at $t = 0$ blue, and $t = T/2$ red, for $T = 2$ on the left and $T = 8$ on the right

will lead to significant deformation of the circle. The the contour line where $\phi = 0.5$ at the maximum deformation is shown in Figure 2 for both values of T.

In Table 1 the results on three different grids are presented for $T = 2$. The accuracy is comparable to the results from Sun and Beckermann [15] who used a the phase-field equation to track the interface, which is similar to our method. The main difference is that using the phase-field equation the reinitialisation and the advection are combined in one equation. Since the conservative level set method separates those two tasks it is possible to use fixed normals for the reinitialisation which leads to the better area conservation compared to the method from [15].

Grid cells	Error	Order	% Area change
32	9.86E-3		1.48
64	2.26E-3	2.1	0.75
128	7.52E-4	1.58	0.71

Table 1: Error and mass loss for vortex test with $T = 2$

If the circle is advected longer in the vortex its develops very small structures and a corresponding resolution is required. As soon as the method is not able to resolve small structures in the interface, it develops small droplets which separate from the main structure as can be seen in Figure 2. This is caused by the fact that the method is conserving the quantity ϕ whereas in an ordinary level set method the unresolved parts simply vanish and therefore cause a mass loss. As the velocity is inverted and the the circle should be recovered at $t = T$ it becomes clear that the small, separated droplets will cause a big distortion of the interface (Figure 3). With a finer mesh the number of droplets which separate decreases and the error at $t = T$ is decreasing. Table 2 summarises the error and the area change for two different grid sizes. The error of the presented method is similar to the error of an ordinary level set method [7] but the area conservation is improved considerably. On the other hand we achieve a lower accuracy and area conservation than the PLS [7].

Grid cells	Error	Order	% Area change
128	1.75E-2		1.47
256	2.19E-3	2.9	0.77

Table 2: Error and mass loss for vortex test with $T = 8$

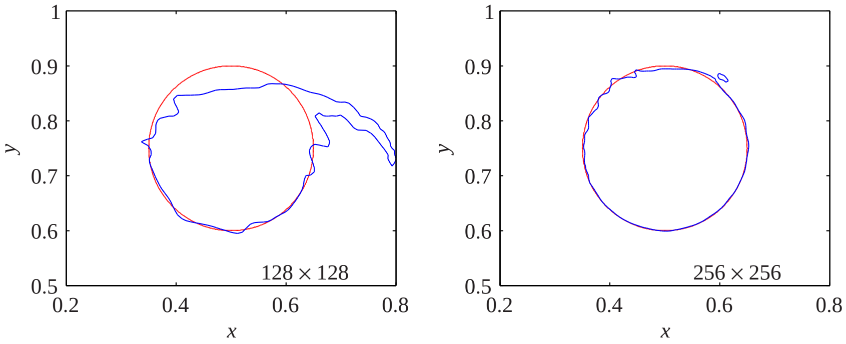

Figure 3: Interface at $t = T = 8$ on a 128×128 and 256×256 grid

3.2 Rigid Body Rotation of Zalesak's Disk

A stream function of

$$\psi(x, y) = -\frac{\pi}{628}\left(x^2 + y^2 - x - y\right) \tag{8}$$

is given in a square unit domain. Initially a slotted circle is placed at $(0.5, 0.75)$. The radius of the circle is 0.15, the width and the length of the slot are 0.05 and 0.25 respectively. In $t = 628$ the slotted disk completes a Rigid Body Rotation around the center of the domain. The time step is set to $\Delta t = 5 \cdot 10^{-3}$ and the width of the hyperbolic tangent is $\epsilon = 0.7\Delta x$.

Figure 4 shows the interface location of interface after one revolution. The errors and area changes are shown in Table 3. Also in this thest the accuracy is slightly better than for phase-field method [15]. The area change is the same for the coarse grid but decreases much faster in our method (0.04% versus 0.8% on the finest grid). If our results

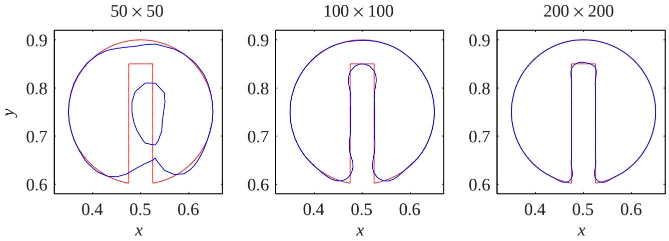

Figure 4: Zalesak's Disk after one revolution on different grids

are compared to an ordinary level set method and a hybrid particle level set method the improvement in mass conservation is evident. Enright et. al. [7] report that on the coarse grid the disk vanishes completely using a level-set method whereas the PLS suffers form a area loss of 15.9%. On the other hand the area loss of the accurate conservative level set method [10] is an order of magnitude smaller than in our method. Note that our errors can not be compared directly with the errors reported in [7] since a different domain size is used and the error measurement in Equation (6) is not dimensionless.

Grid cells	Error	Order	% Area change
50	8.67E-3		3.03
100	1.20E-3	2.85	0.28
200	3.49E-4	1.79	0.04

Table 3: Error and mass loss for one revolution of Zalesak's Disk

4 Conclusion

A finite difference implementation of the conservative level set method has been proposed. The employed discretisation schemes are well documented and widely used, therefore it is relatively easy to implement the conservative level set method into existing flow solvers especially to those which already contain an ordinary level set method. Mass conservation is considerably improved compared to an ordinary level set method and some related methods. Form the results in the test cases it can be seen that our method does not handle very thin interface structures as well as other methods. But this shortcoming can probably be improved by optimising the parameters for the reinitialisation, e.g. the number and frequency of the reinitialisation steps and the width of the hyperbolic tangent.

REFERENCES

[1] Ronald P. Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152(2):457–492, 1999.

[2] Myungjoo Kang, Ronald P. Fedkiw, and Xu-Dong Liu. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, 15(3):323–360, 2000.

[3] Mark Sussman, Peter Smereka, and Stanley Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114(1):146–159, September 1994.

[4] J. A. Sethian and Peter Smereka. Level set methods for fluid interfaces. *Annual Review of Fluid Mechanics*, 35(1):341–372, 2003.

[5] Elin Olsson and Gunilla Kreiss. A conservative level set method for two phase flow. *Journal of Computational Physics*, 210(1):225–246, 2005.

[6] Elin Olsson, Gunilla Kreiss, and Sara Zahedi. A conservative level set method for two phase flow II. *Journal of Computational Physics*, 225(1):785–807, 2007.

[7] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1):83–116, 2002.

[8] Mark Sussman and Elbridge Gerry Puckett. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, 162(2):301–337, August 2000.

[9] Chi-Wang Shu. *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, chapter Essentially Non-oscillatory and Weighted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws, pages 325–432. Springer Berlin / Heidelberg, 1998.

[10] Olivier Desjardins, Vincent Moureau, and Heinz Pitsch. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of Computational Physics*, 227(18):8395–8416, 2008.

[11] J. A. Sethian. Fast marching methods. *SIAM Review*, 41(2):199–235, Jun. 1999.

[12] Kevin T. Chu. LSMLIB. http://ktchu.serendipityresearch.org/software/lsmlib/index.html, 2009.

[13] Bo Strand. Summation by parts for finite difference approximations for d/dx. *Journal of Computational Physics*, 110(1):47–67, 1994.

[14] Rachel Caiden, Ronald P. Fedkiw, and Chris Anderson. A numerical method for two-phase flow consisting of separate compressible and incompressible regions. *Journal of Computational Physics*, 166(1):1–27, 2001.

[15] Y. Sun and C. Beckermann. Sharp interface tracking using the phase-field equation. *Journal of Computational Physics*, 220:626–653, 2007.

Article [b]

Curvature computation for a sharp interface method using the conservative level set method

C. Walker

In A. Eriksson and G. Tibert, editors, *Proceedings of NSCM-23*, pages 203–206, 2010

23rd Nordic Seminar on Computational Mechanics
NSCM-23
A. Eriksson and G. Tibert (Eds)
©KTH, Stockholm, 2010

CURVATURE COMPUTATION FOR A SHARP INTERFACE METHOD USING THE CONSERVATIVE LEVEL SET METHOD

CLAUDIO WALKER*

*Norwegian University of Science and Technology (NTNU)
Department of Energy and Process Engineering
N-7491 Trondheim, Norway
e-mail: claudio.walker@ntnu.no

Key words: Conservative level set, Sharp interface, Curvature

Summary. We compare different approaches for the computation of the curvature at the interface. The first method computes the curvature at the grid points by approximating the derivatives of the conservative level set function. Then the curvature is interpolated to the interface position. The second method uses a distance function which is computed from the conservative level set function by a fast marching method to approximate the derivatives at the grid points. The third method computes the curvature directly at the interface position by fitting a local polynomial through neighboring grid intersections. In our test the method which computes the curvature from approximated derivatives and then interpolates it to the intersections is the most accurate.

1 INTRODUCTION

Computations of two phase flows often use the continuum surface tension approach to handle the surface tension. In order to obtain a smooth solution the jump in the density and viscosity is smeared over multiple grid points around the interface and the singular force resulting from the surface tension is applied as a volume force around the interface. As a result the jump in the pressure is smeared as well. A more accurate description of the jumps at the interface can be obtained using the Ghost Fluid Method (GFM)[1]. It applies the surface tension force directly at the interface and the finite difference stencils are corrected to accommodate the jumps at the interface.

The conventional level set method using a distance function does not conserve the mass of the fluids. To address this problem we use the conservative level set method (CLSM)[2]. The signed distance function is replaced by a hyperbolic tangent function. This allows to write the reinitialization as a conservation law.

The combination of the GFM and the CLSM leads to challenges in the computation of the curvature. Since the surface tension force is applied directly at the interface by the introduction of a jump in the pressure, the curvature has to be computed at the intersections of the interface and the grid lines. These intersection points are obtained by a linear interpolation of the level set function and are therefore second order accurate. To judge whether a method to compute the curvature is applicable to the GFM, the accuracy of the curvature computation has to be assessed at the intersection points.

2 Curvature computation

2.1 Computation using finite difference approximations

The simplest approach to get the curvature at the intersection points is to approximate the derivatives of the conservative level set function at the grid points. That can be done by using e.g. central finite differences. Then the curvature at the grid points is given as

$$\kappa = \frac{\phi_x^2 \phi_{yy} - 2\phi_x \phi_y \phi_{xy} + \phi_y^2 \phi_{xx}}{(\phi_x^2 + \phi_y^2)^{3/2}} \ . \tag{1}$$

This is the standard method to compute the curvature in level set applications. Finally the curvature has to be interpolated to the intersection points which can be done by the same linear interpolation method used to find the intersection points.

2.2 Least squares approximation from distance function

If the conservative level set function is advected using certain discretisation schemes, it will develop spurious oscillations, which will lead to problems for the approximation of the interface normals and curvature. To avoid these problems Desjardins et al.[3] propose to first recompute a signed distance function ψ from the conservative level set function ϕ using a fast marching method (FMM). Then the derivatives in equation (1) are approximated from ψ. Because the FMM is at most second order accurate, the curvature will not converge if the derivatives are computed with finite differences. By using the least squares approach[4] first order convergence is observed.

2.3 Curve fitting through the intersection points

Since all intersection points of the interface with the grid lines are known we can use those to compute the curvature directly. Suppose there is a parameterised curve $\vec{x}(s) = (x(s), y(s))$ trough the intersection points. The curvature at a intersection point is then given by

$$\kappa = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{3/2}} \ , \tag{2}$$

where $\dot{x}$ and $\ddot{x}$ first and second derivatives, respectively, with respect to the parameter s. To approximate the derivatives at an intersection point $\vec{x}_0 = (x_0, y_0)$ we use two additional intersection points $\vec{x}_m$ and $\vec{x}_p$ which are located before and after $\vec{x}_0$ on the interface, respectively. Assuming that the difference of the parameter s is the same on both sides we can use simple finite difference stencils to compute the derivatives, e.g. $\dot{x}_0 = (x_p - x_m)/2$.

It was only possible to obtain converging curvature estimates if the linear interpolation of the intersection points was replaced by a more accurate cubic interpolation. We got the best results by setting $\vec{x}_m$ and $\vec{x}_p$ to the neighbouring points where the interface intersects a parallel grid line to the one in $\vec{x}_0$. At certain points where two intersection points are located extremely close to a grid point this approach will lead to large errors and the curvature will not converge as the grid is refined. To find a remedy the symmetry factor q of each curvature approximation stencil is computed as

$$q = \frac{\delta_m + \delta_p - 2\sqrt{\delta_m \delta_p}}{2\sqrt{\delta_m \delta_p}} \,, \tag{3}$$

where $\delta_m = |\vec{x}_m - \vec{x}_0|$ and $\delta_p = |\vec{x}_p - \vec{x}_0|$ respectively. q will be zero if the stencil is perfectly symmetric. If the symmetry factor of the approximation in a certain intersection point is 10 times higher than the average of his two neighbours, the curvature at this point will be linearly interpolated from the adjacent points which have a more symmetric stencil.

3 Numerical test

To test the presented methods for the curvature, we place a circle with radius $r = 1/3$ into the center of a square domain of size $[-1, 1] \times [-1, 1]$ with n grid points in each direction. We initialise the conservative level set function with

$$\phi = \left(1 + e^{d/\epsilon}\right)^{-1} \,, \tag{4}$$

where $\epsilon = 0.2\sqrt{2/n}$ is the parameter for the slope of ϕ at the interface, and d is the analytical distance form the interface. Then the reinitialisation equation is solved to steady state to ensure that we test the curvature comptation on a level set function which can be expected in a two phase solver. This reinitialized conservative level set function is then used to compute the curvature κ at each intersection point. To test the least squares approximations from the distance function (section 2.2) we used the FMM from the LSMLIB [5] with second order upwinding to obtain ψ from d. The error of the curvature is then calculated by

$$e = \left(\frac{1}{m}\sum_{i=1}^{m}(\kappa_i - 3)^2\right)^{1/2} \tag{5}$$

where m is the number of intersection points.

The results are presented in Figure 1. If the finite difference method (section 2.1) is used the error in the curvature reduces with second order, which is consistent with both the finite difference and the interpolation schemes. The curve fitting at the intersection points (section 2.3) leads to a convergence which is between first and second order. At a low resolution 2nd order can be observed. The lower convergence rate at higher resolution stems from the high frequency oscillations which are illustrated on the right of Figure 1. Those oscillations originate from the variance in the stencil quality for each intersection point. The error could be further reduced by low pass filtering the curvature along the interface. The curvature which is computed from ψ using a least squares approximation of the derivatives (section 2.2) is the most accurate one at the lowest resolution, but it does not converge as the grid is refined. The reason for this behavior is that ψ which is computed with the FMM is only second order accurate at the lowest resolution. As the grid is refined the accuracy of the FMM deteriorates to first order.

4 Conclusions

We tested alternative methods to compute the curvature from a conservative level set function. If the curvature is computed by fitting a curve through the intersection points of the

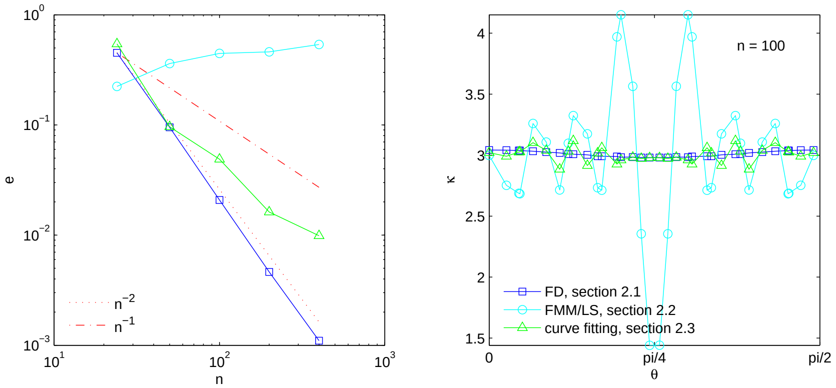

Figure 1: Error of the curvature computation on the left, curvature for $n = 100$ on the right

interface with the gird lines, the intersection points have to be interpolated with at least cubic interpolation to obtain convergent results. The method which fits a curve through the intersection points is less accurate than the method using finite differences, but it converges faster than the least square approximation of the derivatives of a distance function computed by the FMM. In our tests the least squares approximation is not consistent with grid refinement. If an accurate curvature is important, one should try to keep the conservative level set function free from oscillations, e.g. by using WENO schemes for the advection and frequent reinitialisation, such that the curvature can be computed by finite difference from the conservative level set function.

REFERENCES

[1] Kang, M., Fedkiw, R. P. & Liu, X.-D. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing* **15**, 323–360 (2000). URL `http://www.springerlink.com/content/w123625121021441`.

[2] Olsson, E., Kreiss, G. & Zahedi, S. A conservative level set method for two phase flow II. *Journal of Computational Physics* **225**, 785–807 (2007). URL `http://www.sciencedirect.com/science/article/B6WHY-4MYMG2M-1/2/cafe93c2625800b464f14fdd52981be8`.

[3] Desjardins, O., Moureau, V. & Pitsch, H. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of Computational Physics* **227**, 8395–8416 (2008). URL `http://www.sciencedirect.com/science/article/B6WHY-4SRCJWK-1/2/a8c8127ed57af20851bac3a9f75076c9`.

[4] Marchandise, E., Geuzaine, P., Chevaugeon, N. & Remacle, J.-F. A stabilized finite element method using a discontinuous level set approach for the computation of bubble dynamics. *Journal of Computational Physics* **225**, 949 – 974

(2007). URL http://www.sciencedirect.com/science/article/B6WHY-4MV0MDV-3/2/40c95b8384130d1b080e722c78946327.

[5] Chu, K. T. Lsmlib. Website (2009). URL http://ktchu.serendipityresearch.org/software/lsmlib/index.html.

Article [c]

Contact line treatment with the sharp interface method

C. Walker and B. Müller

In B. Skallerud and H. Andersson, editors, *MekIT'11: Sixth national conference on Computational Mechanics*, pages 451–462. Tapir Academic Press Trondheim, 2011

Contact line treatment with the sharp interface method

Claudio Walker* **and Bernhard Müller**
Department of Energy and Process Engineering
The Norwegian University of Science and Technology, Trondheim, Norway
e–mail: claudio.walker@ntnu.no

Summary In the present study a method to handle contact points in a sharp interface method is presented. The contact points are tracked explicitly and their velocities are a function of the contact angle. For sharp interface methods an accurate approximation of the curvature is important. This cannot be obtained by using the conventional central difference methods adjacent to walls since the level set functions are not defined therein. Therefore we propose to use the tracked contact point and the intersection points of the first two grid lines parallel to the wall with the zero contour line of the signed distance function to approximate the contact angle and curvature adjacent to the walls.

Introduction

Drop impacts on dry surfaces can be found in many industrial and natural processes. Applications where the behavior of the impacting drops plays an important role include ink-jet printing, spray cooling, pesticide spraying, erosion processes due to rain and thermal spray coating. Probably the first study on droplet impacts was performed by [24]. An overview on the different phenomena which can take place during a drop impact is given in the reviews of [17] and [25].

One of the difficulties arising in modelling droplet impacts is the moving contact line problem. If the conventional no-slip boundary condition is applied to the Navier-Stokes equations, the stresses are diverging at the line where the three phases meet. In fact, molecular dynamics (MD) simulations show a near complete slip in the region of the contact line [10]. Although there are several empirical models connecting the dynamic contact angle at the contact line with the velocity of the contact line, a complete understanding of the mechanics at the contact line is still missing.

The surface tension plays an important role during the spreading and receding of an impacting drop. Therefore, accurate modelling of the surface tension in the continuum model is important. The surface tension introduces a jump in the solution across the interface. While this jump is smeared out to obtain numerically smooth solutions in diffuse interface methods, the jump conditions are imposed by sharp interface methods. The ghost-fluid method (GFM) [4], which uses a fixed Cartesian grid, was extended to incompressible two-phase flows [9]. The GFM allows to retain the jumps across the interface and therefore it is possible to eliminate the error which stems from the artificial smearing of the fluid properties. To use the GFM it is necessary to know the location of the interface, as well as its curvature. This information about the interface geometry can be obtained from the the level-set approach [23] [15]. However, if the interface crosses a wall boundary, i.e., if it forms a contact point, a special method to approximate the geometric information is required. Spelt [22] proposed a method to treat contact points in diffuse interface methods, where contact points are tracked explicitly. In the presented paper we present an extension of Spelt's method to a sharp interface method.

Equations

Navier-Stokes equations

We consider incompressible flow of two immiscible viscous fluids. In this study we confine ourselves to two-dimensional problems. There is no fundamental obstacle to extend the presented method to three dimensions. The continuity and momentum equations, i.e., the Navier-Stokes equations for a Newtonian fluid for incompressible flow:

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{g} \tag{2}$$

where $\mathbf{u}$ is the velocity vector , p is the pressure, $\mathbf{g}$ is the gravity vector, μ and ρ are the dynamic viscosity and the density, respectively. The material properties μ and ρ can be different in each fluid and we use a $+$ and $-$ sign to discriminate between the two fluids. These equations have to be fulfilled in both fluids in two-phase flow.

Interface conditions

The boundary conditions at the interface can be derived by considering an infinitesimal control volume across the interface. This volume can be divided into a control volume for each fluid. The conservation laws for mass and momentum hold for the entire volume as well as for each of the partial volumes. Subtracting the sum of the conservation laws applied to the two partial volumes individually from the conservation laws applied to the entire volume, we get the boundary conditions at the interface. For the case with constant surface tension σ and no mass transfer across the interface we get the following jump conditions:

$$[\mathbf{u}] = 0 \tag{3}$$

$$\left[\begin{pmatrix} \mathbf{n}^T \\ \mathbf{t}^T \end{pmatrix} (p\mathbf{I} - \tau)\, \mathbf{n} \right] = \begin{pmatrix} \sigma\kappa \\ 0 \end{pmatrix}, \tag{4}$$

where square brackets define the jump across the interface, e.g. $[\mathbf{u}] = \mathbf{u}^+ - \mathbf{u}^-$. We further denote $\mathbf{n}$ and $\mathbf{t}$ as unit normal and tangent vectors to the interface. κ is the local interface curvature and τ is the viscous stress tensor. These conditions imply that the velocity and tangential stresses are continuous across the interface, whereas the pressure and the normal stresses are discontinuous.

Level set method

In order to apply the interface conditions presented in the previous section it is necessary to know its position. A popular method to keep track of the interface position in two- phase flows is the level set method (LSM) [19]. There the interface is defined as the zero contour line of a scalar function ϕ. Typically ϕ is the signed distance function from the interface and it exists and is continuous through the entire computation domain. The signed distance function is advected with the local fluid velocity using the advection equation

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0. \tag{5}$$

Since all discretisations of the advection equation will not be exact, ϕ looses its signed distance property over time and has to be reinitialised solving the following equation to steady state

$$\frac{\partial \phi}{\partial \tau} + \operatorname{sign}(\phi)\left(|\nabla \phi| - 1\right) = 0. \tag{6}$$

The interface normal and curvature can be obtained directly from the signed distance function.

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|} \tag{7}$$

$$\kappa = -\nabla \cdot \mathbf{n} \tag{8}$$

Contact point

Around a point where the interface between two fluids meets a solid surface, the conventional no-slip boundary condition cannot be applied. For otherwise the stresses around the interface would become singular. To avoid this singularity the no-slip boundary condition is often replaced by a slip boundary condition of the following form:

$$u_{\|} = \lambda \left. \frac{\partial u_{\|}}{\partial x_{\perp}} \right|_{\text{wall}}, \tag{9}$$

where λ is the slip length and $\|$ and $\perp$ denote the parallel and normal directions relative to the wall. The difference in the surface energies can lead to a motion of the contact point. For small capillary numbers this effect can become extremely important. However, it is not possible to describe the dynamics around the contact point accurately using the Navier-Stokes equations. Some authors choose to fix the contact angle to model the dynamics around the contact line [1, 2, 5, 12]. In the present paper we shall set the contact point velocity u_{CP} as a function of the contact angle. To achieve this goal Spelt [22] proposed to track the contact point explicitly. Thus, the position of the contact point is described by an ordinary differential equation

$$\frac{\mathrm{d}x_{\text{CP}}}{\mathrm{d}t} = u_{\text{CP}} = f(\theta), \tag{10}$$

where $f(\theta)$ is an arbitrary function describing the dependency of the contact point velocity on the contact angle. The choice of f is important and should be done carefully to obtain good results. Different approaches to model the dependency of the contact point velocity on the contact angle are possible, including the use of empirical data or results from microscale simulations around the interface. In the present paper we uses a simple linear relation as an example.

Discretisation

The discretisation is done on a uniform staggered grid, where the scalar quantities, i.e., pressure p and signed distance function ϕ, are stored at the cell centers while the velocity components u and v are stored at the vertical and horizontal cell faces, respectively [7]. The classical marker and cell (MAC) method is used to couple velocity and pressure in the discretisation of the incompressible Navier-Stokes equations (1) and (2).

Navier-Stokes

The advection terms in the Navier-Stokes equation (2) are discretised by a 5th order finite difference WENO scheme using a Lax-Friedrichs flux splitting [20]. For the WENO scheme can handle the discontinuities in the first derrivative of the velocity automatically. In points which are not adjacent to the interface the viscous terms are discretised by second order central difference stencils, i.e.,

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i-1/2,j} \approx \frac{u_{i+1/2,j} - 2u_{i-1/2,j} + u_{i-3/2,j}}{\Delta x^2} \tag{11}$$

$$\left(\frac{\partial^2 u}{\partial y^2}\right)_{i-1/2,j} \approx \frac{u_{i-1/2,j+1} - 2u_{i-1/2,j} + u_{i-1/2,j-1}}{\Delta y^2}. \tag{12}$$

The velocity component in y-direction is treated analogously. To enforce incompressibility a direct projection is applied. First an intermediate velocity $\mathbf{u}^*$ is obtained by updating the velocity from the previous time step with the advective, viscous and gravity terms.

$$\mathbf{u}^* = \mathbf{u} + \Delta t \left(-\left(\mathbf{u}\cdot\nabla\right)\mathbf{u} + \frac{\mu}{\rho}\nabla^2\mathbf{u} + \mathbf{g}\right) \tag{13}$$

This intermediate velocity field is then used as the right hand side of the Poisson equation for the scaled pressure correction $p^* = p/\Delta t$. Homogeneous Neumann boundary conditions are applied to solve for p^*.

$$\nabla \cdot \left(\frac{\nabla p^*}{\rho}\right) = \nabla \cdot \mathbf{u}^* \tag{14}$$

Away from the interfaces the density ρ corresponds to the constant density of the fluid we are in and the Laplace operator is approximated by the standard second order 5 point central finite difference stencil. Finally the intermediate velocity is made divergence free using the solution of the pressure Poisson equation.

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\nabla p^*}{\rho} \tag{15}$$

This projection procedure can be viewed as a special time splitting scheme which is advancing the solution Δt in time like one time step in a forward Euler scheme. Therefore a repetition of the projection procedure can be employed to form a Runge-Kutta time integration scheme. In the present work the 3rd order TVD Runge-Kutta method by Shu and Osher [21] is applied. An expression for the maximum allowed time step for a forward Euler time integration considering advection, diffusion, surface tension and gravity is given by Kang et al. [9]:

$$\frac{\Delta t}{2}\left((C_{\mathrm{CFL}} + V_{\mathrm{CFL}}) + \sqrt{(C_{\mathrm{CFL}} + V_{\mathrm{CFL}})^2 + 4S_{\mathrm{CFL}}^2}\right) \le 1, \tag{16}$$

where $C_{\mathrm{CFL}} = \frac{|u|_{\max}}{\Delta x} + \frac{|v|_{\max}}{\Delta x}$, $V_{\mathrm{CFL}} = \max\left\{\frac{\mu^-}{\rho^-}, \frac{\mu^+}{\rho^+}\right\}\frac{4}{\Delta x^2}$ and $S_{\mathrm{CFL}} = \sqrt{\frac{\sigma|\kappa|}{\min\{\rho^-,\rho^+\}\Delta x^2}}$.

Interface jump conditions

The jump conditions at the interface (4) can be rewritten such that jumps are separated for the derivatives of the velocity components [9].

$$\begin{aligned}\begin{pmatrix}[\mu u_x] & [\mu u_y]\\ [\mu v_x] & [\mu v_y]\end{pmatrix} =& [\mu]\begin{pmatrix}(\nabla u)^T\\(\nabla v)^T\end{pmatrix}\begin{pmatrix}0\\ \mathbf{t}^T\end{pmatrix}^T\begin{pmatrix}0\\ \mathbf{t}^T\end{pmatrix} + [\mu]\mathbf{n}\mathbf{n}^T\begin{pmatrix}(\nabla u)^T\\(\nabla v)^T\end{pmatrix}\mathbf{n}\mathbf{n}^T\\ &- [\mu]\begin{pmatrix}0\\ \mathbf{t}^T\end{pmatrix}^T\begin{pmatrix}0\\ \mathbf{t}^T\end{pmatrix}\begin{pmatrix}(\nabla u)^T\\(\nabla v)^T\end{pmatrix}^T\mathbf{n}\mathbf{n}^T\end{aligned} \tag{17}$$

$$[p^*] = \Delta t\left(\sigma\kappa + 2[\mu]\begin{pmatrix}\nabla u\cdot\mathbf{n}\\ \nabla v\cdot\mathbf{n}\end{pmatrix}\cdot\mathbf{n}\right) \tag{18}$$

The jumps are continuous functions which are defined in the whole domain. Therefore they can be computed at the grid centers and then interpolated to the location where they are required. To compute the jump conditions one needs to approximate the derivatives of the velocity components at the cell centers. Kang et al. [9] choose to first interpolate the velocity components to the cell centers and then use standard central differences to approximate the gradients of the velocity. Here a different strategy is adopted, where the derivatives u_x and v_y are directly computed as the difference approximations of the velocities at the cell faces, i.e., $u_{x\,i,j} = \left(u_{i+1/2,j} - u_{i-1/2,j}\right)/\Delta x$, and the other two derivatives namely u_y and v_x are approximated at the cell corners, i.e., $u_{y\,i+1/2,j+1/2} = \left(u_{i+1/2,j+1} - u_{i+1/2,j}\right)/\Delta y$ and those two derivatives are subsequently interpolated to the cell centers. Both methods lead to the same stencil for u_y and v_x, but for u_x and v_y the direct approximation results in a smaller stencil with the same order of accuracy. In addition we point out that u_x and v_y at the cell centers and u_y and v_x at the cell corners have to be computed anyway to approximate the viscous terms, cf. equations (11) and (12).

The separated jump conditions together with the ghost-fluid method [13] allow to treat the jumps in a sharp manner. If there is an interface crossing the grid lines of a stencil, the known jumps are added or subtracted from the points on the opposite side of the interface. This addition or subtraction is then included into the second order finite difference method. The treatment of the jump condition is improved by interpolating the jump condition to the place where the interface intersects with the grid lines. This procedure allows to discretise the Poisson equation

$$(\beta\varphi_x)_x + (\beta\varphi_x)_x = f(x) \tag{19}$$

with the interface conditions $[\varphi] = a$ and $[\beta\varphi_n] = b$ in the following way:

$$\begin{aligned}&\frac{\beta_{i+1/2,j}\left(\frac{\varphi_{i+1,j}-\varphi_{i,j}}{\Delta x}\right) - \beta_{i-1/2,j}\left(\frac{\varphi_{i,j}-\varphi_{i-1,j}}{\Delta x}\right)}{\Delta x}\\ &+\frac{\beta_{i,j+1/2}\left(\frac{\varphi_{i,j+1}-\varphi_{i,j}}{\Delta y}\right) - \beta_{i,j-1/2}\left(\frac{\varphi_{i,j}-\varphi_{i,j-1}}{\Delta y}\right)}{\Delta y} = f_{i,j} + F.\end{aligned} \tag{20}$$

F is a correction term for the jumps and therefore dependent on the interface position and the jump conditions a and b. If no interface crosses the stencil, F is zero and equation (20) reduces to the conventional second order central difference method. The correction appears only on the right hand side, which is beneficial since the same linear solvers can be used as for conventional central difference discertisations. If for example the interface intersects the left arm of the stencil, i.e., it is located between $x_{i-1,j}$ and $x_{i,j}$ and at $x_{i-1,j}$ we have the fluid with the material constant β^+, the correction term reads:

$$F = \frac{\beta_{i-1/2,j}a_\Gamma}{\Delta x^2} - \frac{\beta_{i-1/2,j}b_\Gamma\Theta}{\beta^+\Delta x}, \tag{21}$$

where $\Theta \cdot \Delta x$ is the distance between $x_{i-1,j}$ and the intersection point, and the subscript Γ indicates that the jump conditions are interpolated to the intersection point. In the context of two-phase flow this method is used for the viscous terms, where φ corresponds to the velocity components, $a = 0$ and b are the jump conditions form equation (17). For the pressure Poisson equation φ corresponds to the pressure p^* and the derivative is continuous, i.e., $b = 0$ and the pressure jump a is given by equation (18). For details concerning the GFM and its implementation for two-phase flows we refer the reader to the literature [9, 13].

Level set method

The gradients in the level set equations (5) and (6) are discretised by a 5th order finite difference WENO scheme [14]. To minimise spurious displacements of the interface during the reinitialisation the constrained reinitialisation CR-1 by Hartmann et al. [8] is applied. The normal is approximated by a conventional second order finite difference scheme. The expression for the curvature (8) can be rewritten as:

$$\kappa = -\frac{\phi_x^2 \phi_{yy} - 2\phi_x \phi_y \phi_{xy} + \phi_y^2 \phi_{xx}}{\left(\phi_x^2 + \phi_y^2\right)^{3/2}}, \tag{22}$$

where the derivatives are approximated by second order finite differences as well.

Contact point treatment

The contact point is tracked explicitly with the ordinary differential equation (10). Immediately after the advection of ϕ the contact point is updated using the velocity which was computed form the contact angle θ at the last time step.

$$x_{\text{CP}}^{n+1} = x_{\text{CP}}^{n} + \Delta t \cdot f(\theta^n) \tag{23}$$

Since the updating of the contact point is a part of each Euler step in the projection algorithm, which forms one step of the Runge-Kutta scheme, the order of the temporal discretisation of the contact point position is consistent with the temporal order of the other equations.

The curvature κ and the contact angle cannot be computed with central differences at the wall since the stencil for the first grid point inside the fluid would contain points which are located inside the wall. As discussed in more detail below, the interface is assumed to continue as a straight line into the wall and the level set values at the ghost points are set as signed distances to that line. As a result the curvature at the first grid point would be compromised if computed by central differences. Instead a circle is fitted through the contact point x_{CP} and the intersection points of the interface with the first two grid lines parallel to the wall. To get convergent curvatures the intersection points of the grid lines and the zero level set contour are determined by a quadratic interpolation. The curvature at the first gird point is then given as the inverse of the radius R from the fitted circle. The smaller angle between the wall and the fitted circle is:

$$\hat{\theta} = \sin^{-1}\left(\frac{|x_{\text{CP}} - x_m|}{R}\right) \tag{24}$$

where x_m is the x-position of the center from the fitted circle, cf. Fig. 1. Using the y-position of the circle center and the sign of the first grid point left of the interface it can be checked whether $\hat{\theta}$ is the contact angle in the desired fluid and we set $\theta = \hat{\theta}$. Otherwise the correct angle is obtained by $\theta = \pi - \hat{\theta}$. In Figure 1 the fitted circle at the contact point is illustrated. During the reinitialisation the interface suffers from spurious displacement. Therefore the reinitialisation will change the measured contact angle and curvature since they depend on the first two interpolated intersection points. The spurious displacement and therefore the change in the contact angle and curvature can efficiently be reduced using a constrained reinitialisation.

The reinitialisation equation (6) can be rewritten as a hyperbolic equation where the characteristics are pointing perpendicular away form the interface [18]. Therefore the position of the interface decides where a boundary condition for the reinitialisation is required. In areas where no boundary condition is required the ghost points for the signed distance function can simply be filled with a linear extrapolation from the fluid points. To fill the ghost points in areas where a boundary condition is required, the interface is prolonged as a straight line from the contact point with a slope given by θ. For all ghost points which are below a line which is perpendicular to the interface at the contact point (shaded area in Figure 1) the distance to the prolonged interface is computed analytically. In addition the sign of the distance can be decided using the first fluid point adjacent to the interface. To ensure that the characteristics are obeyed the ghost points are finally filled with either the extrapolated value or the analytical signed distance whichever has the smallest absolute value. If there are multiple contact points the analytically determined distance to the closest prolonged interface is used.

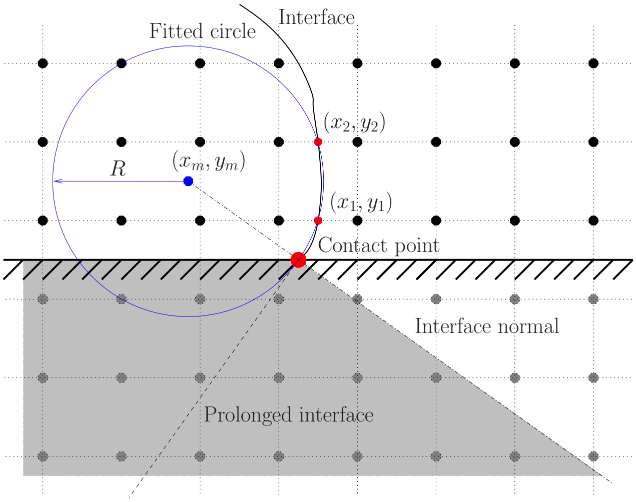

Figure 1: Illustration of the contact point treatment, in the shaded region the ghost points are filled with the analytical distance to the interface if this is smaller than the extrapolated value.

Results

To test our implementation of the GFM we first compute two examples of two-phase flow with dynamic interfaces for which analytical solutions exist. The ability of our method to handle contact points is demonstrated in an additional example of a droplet receding on a solid wall.

Oscillating droplet

An oscillating droplet surrounded by another fluid where both fluids are inviscid is a popular test case for multiphase solvers. An analytical solution to this problem can be found using potential theory [11]. Initially the droplet has the shape of an ellipse with a semimajor and a semiminor axis which is 5% longer and smaller than the radius $r = \frac{1}{3}$.

The surface tension $\sigma = 1$ N/m and the density $\rho^+ = \rho^- = 1$ kg/m^3 will lead to an oscillation which is not damped in the inviscid case. The frequency of the oscillation is $\omega_0^2 = \frac{6\sigma}{(\rho^+ + \rho^-) r_0^3}$. Hansen [6] showed that the influence of the no-slip boundaries are becoming small if the domain becomes larger than 2×2 m^2. This was also verified in the present case. Therefore all the reported results were computed with a domain size of 2×2 m^2. The time step was chosen such that it is 1% of the maximum allowed time step and the number of grid points was $m = 52$, $m = 100$ and $m = 200$.

The time evolution of the semimajor axis is plotted in Figure 2. As the grid is refined the numerical solution approaches the exact solution. The amplitude of the semimajor axis is damped in the numerical results and the frequency is slightly too low as well. To analyse the influence of the time step we computed a solution on the coarsest grid with a 10 times larger Δt. It can be seen in Figure 2 that the results are converging with decreasing time step. The area loss after two oscillation periods is 0.356%, 0.077% and 0.004% for the three grid resolutions. This is comparable to the results by Hansen [6].

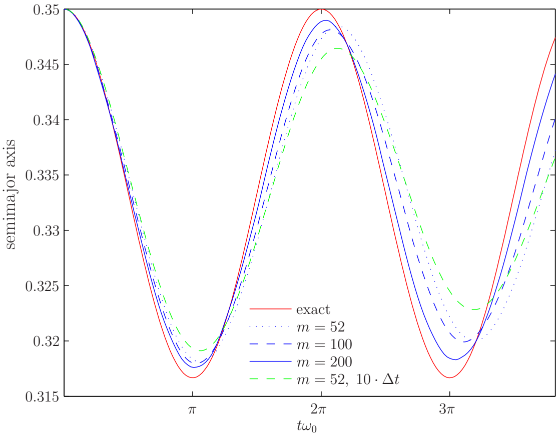

Figure 2: Time evolution of semimajor axis for the oscillating droplet.

Standing wave

Viscous damping of surface waves is used to verify the two phase solver for cases where viscous and surface tension forces interact. An analytical solution for this problem was presented by Prosperetti [16]. We use non-dimensional variables here. Initially two fluids are separated by a flat interface in the middle of the domain which is slightly perpetuated by a sine wave. The wavelength of the perturbation is $\lambda = 2\pi$ and the initial amplitude is set to $A_0 = 0.01\lambda$. According to the analytical solution the oscillation frequency is $\omega_0^2 = \frac{\sigma}{\rho^+ + \rho^-}$.

All solutions are obtained in a domain of size $2\pi \times 2\pi$. The boundary conditions parallel to the interface are slip walls, and perpendicular to the interface periodicity is enforced. Both densities are set to unity, the surface tension is $\sigma = 2$ and the viscosities are set to $\mu = 0.064720863$. For all reported grid resolutions the time step was set to $\Delta t = 0.01$.

Figure 3 shows the time evolution of the amplitude and the amplitude error. Again the frequency predicted by the numerical computations is too low, but converges with grid refinement. Opposed to the inviscid droplet oscillation, the damping of the oscillation amplitude is not strong enough. The RMS values of the amplitude errors are summarised in Table 1. A convergence rate between first and second order is observed. The RMS errors are slightly higher than those reported by Desjardins et al. [3]. The lower accuracy in our method could be caused by the completely explicit time integration, whereas a semi-implicit Crank-Nicholson scheme is employed by Desjardins et al.

Droplet on plate

Our last example is a droplet on a plate receding to its static contact angle. Again, non-dimensional variables are used. A droplet of radius $r_0 = 1.66$ and contact angle of 30° is placed on a solid surface. A different fluid is surrounding the droplet. The static contact angle is $\theta_s = 120°$. This

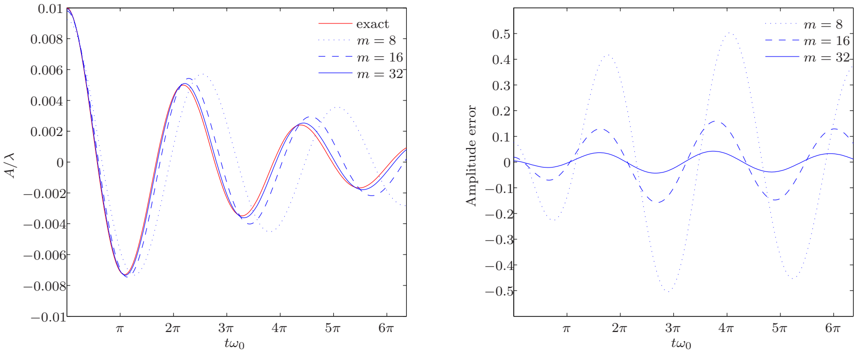

Figure 3: Time evolution of the wave amplitude (left) and amplitude error (right).

Table 1: RMS of amplitude error for the standing wave

Mesh	Error
8×8	0.2962
16×16	0.0958
32×32	0.0261

configuration will lead to moving contact points until the static contact angle is reached. In the steady state all velocities should vanish and the interface will assume a circular shape.

A rectangular computational domain of size $[-1, 1] \times [0, 1]$ is used. All boundaries are no-slip walls except for the bottom wall where the drop is placed on, here a slip boundary condition, cf. equation (9) with a slip length of $\lambda = 0.02$ is applied. The fluid parameters are chosen as in Case II of Spelt [22], i.e., $\rho^- = 1$, $\rho^+ = 20$, $\mu^- = \mu^+ = 0.0495$ and $\sigma = 2.21$, where the fluid denoted with $-$ is inside the droplet. The function of the contact point velocity is $u_{\text{CP}} = 0.1 \cdot (\theta - \theta_s)$. A time step of 0.7 times the maximum stable time step is chosen for all grid resolutions.

The velocity field at $t = 2$ is shown in Figure 4. During the receding of the contact points two vortices develop. The maximum absolute value of the velocity is always located around the contact points and the slip allowed from the boundary condition is clearly visible. In Figure 5 the time evolution of the interface is plotted. Since the Capillary number of the flow is relatively low $\text{Ca} = \frac{\mu u_{\text{CP}}}{\sigma} \leq 3.528 \cdot 10^{-3}$, the interface keeps a circular shape during the entire receding process. The evolution of the contact angle θ and the position of the right contact point x_{CP} as a function of time are presented in Figure 6. The contact angle and contact point position are almost identical for both grid resolutions. Initially the contact angle is small resulting in a high contact point velocity. This is causing a relative large curvature at the contact point which in turn leads to a relatively large pressure jump adjacent to the contact point. Therefore high slip velocities are produced. As the contact points are receding, the contact angle gets smaller and therefore the velocities are reduced as well. Finally the contact points approach their equilibrium positions, the vortices are dissipated and steady state is reached.

At steady state the interface position error is computed by interpolating the position of all intersection points between the grid lines and the zero contour line using a 1-dimensional cubic spline interpolation. The position error of the interface $\varepsilon_{\text{interface}}$ is then given as the 2-norm of the distance between each interpolated intersection point and its correct position. The relative error of the contact point at steady state ε_{CP} is given as the difference between the contact point and its analytically position. Table 2 summarises the errors when the steady state is reached ($t = 24$). Since the interface is slightly displaced during reinitialisation of the signed distance function the curvature at the contact point is changed as well. This causes small spurious velocities around the contact point. The reported norms of the velocity are computed directly after reinitialisation, they will decrease until the next reinitialisation. Tests showed that they will converge to zero if no reinitialisation is applied after the steady state is reached. An important contribution to the errors of the interface and contact positions is the mass loss which occurs during the computation. The interface approaches the shape of a circle but with a radius which is too small. Often the signed distance function is lifted after reinitialisation to eliminate the mass loss. But this procedure would lead again to a spurious change in the curvature at the contact point. Therefore we did not adopt this practice in the presented examples.

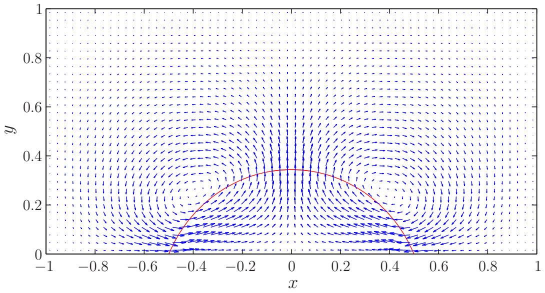

Figure 4: Velocity field at $t = 2$, the largest velocity 0.087 appears at the contact points.

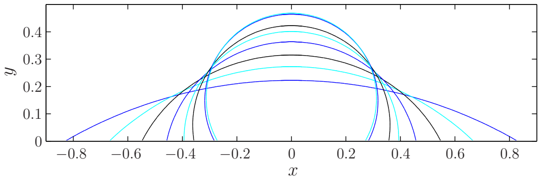

Figure 5: Interface at $t = 0$, 1, 2, 3, 4, 5, 10 and 20.

Conclusions

We devise a way to handle contact points in sharp interface methods. The main difficulty is the computation of the curvature at the first grid point adjacent to the wall. A method is presented where a contact point is tracked explicitly. A circle is fitted through this contact point and the intersection points of the zero contour line of the signed distance function with the first two

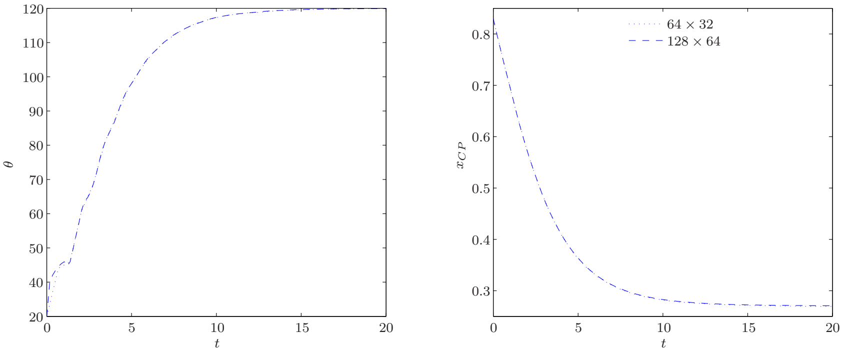

Figure 6: Time evolution of the contact angle (left) and the contact point position (right).

Table 2: Errors at $t = 24$ for a droplet on a plate.

	$\|u\|_2$	$\|v\|_2$	$\varepsilon_{\mathrm{CP}}$	$\varepsilon_{\mathrm{interface}}$
64×32	$0.719 \cdot 10^{-3}$	$0.858 \cdot 10^{-3}$	$13.359 \cdot 10^{-3}$	$5.320 \cdot 10^{-3}$
128×64	$0.391 \cdot 10^{-3}$	$0.381 \cdot 10^{-3}$	$7.903 \cdot 10^{-3}$	$3.353 \cdot 10^{-3}$

grid lines parallel to the wall is used to calculate the contact angle and the curvature adjacent to the wall. Numerical experiments show that the present two-phase solver based on the sharp interface method can correctly simulate dynamic interfaces and the steady state of contact lines for standard test cases, although an improvement is still possible.

Acknowledgements

The authors thank Martin Kronbichler and Prof. Gunilla Kreiss, Department of Information Technology, Uppsala University, Sweden, for the fruitful discussions on contact line dynamics.

References

[1] M.Bussmann, S.Chandra and J.Mostaghimi Modeling the splash of a droplet impacting a solid surface *Physics of Fluids*, **vol.12**(12), 3121–3132, 2000.

[2] D.Caviezel, C.Narayanan and D.Lakehal Adherence and bouncing of liquid droplets impacting on dry surfaces *Microfluidics and Nanofluidics*, **vol.5**, 469 – 478, 2008.

[3] O.Desjardins, V.Moureau and H.Pitsch An accurate conservative level set/ghost fluid method for simulating turbulent atomization *Journal of Computational Physics*, **vol.227**(18), 8395–8416, 2008.

[4] R. P.Fedkiw, T.Aslam, B.Merriman and S.Osher A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method) *Journal of Computational Physics*, **vol.152**(2), 457–492, 1999.

[5] P. R.Gunjal, V. V.Ranade and R. V.Chaudhari Dynamics of drop impact on solid surface: Experiments and vof simulations *AIChE Journal*, **vol.51**(1), 59–78, 2005.

[6] E. B.Hansen *Numerical Simulation of Droplet Dynamics in the Presence of an Electric Field* PhD thesis, Norwegian University of Science and Technology, 2005.

[7] F. H.Harlow and J. E.Welch Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface *Physics of Fluids*, **vol.8**(8), 2182, 1965.

[8] D.Hartmann, M.Meinke and W.Schröder The constrained reinitialization equation for level set methods *Journal of Computational Physics*, **vol.229**(5), 1514 – 1535, 2010.

[9] M.Kang, R. P.Fedkiw and X.-D.Liu A boundary condition capturing method for multiphase incompressible flow *Journal of Scientific Computing*, **vol.15**(3), 323–360, 2000.

[10] J.Koplik, J. R.Banavar and J. F.Willemsen Molecular dynamics of fluid flow at solid surfaces *Physics of Fluids A: Fluid Dynamics*, **vol.1**(5), 781–794, 1989.

[11] H.Lamb *Hydrodynamics* Cambridge University Press, 1932.

[12] H.Liu, S.Krishnan, S.Marella and H.Udaykumar Sharp interface Cartesian grid method II: A technique for simulating droplet interactions with surfaces of arbitrary shape *Journal of Computational Physics*, **vol.210**(1), 32–54, November 2005.

[13] 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*, **vol.160**(1), 151–178, May 2000.

[14] X.-D.Liu, S.Osher and T.Chan Weighted essentially non-oscillatory schemes *Journal of Computational Physics*, **vol.115**(1), 200 – 212, 1994.

[15] S.Osher and R.Fedkiw *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied Mathematical Sciences* Springer-Verlag New York, Inc., 2003.

[16] A.Prosperetti Motion of two superposed viscous fluids *Physics of Fluids*, **vol.24**(7), 1981.

[17] M.Rein Phenomena of liquid drop impact on solid and liquid surfaces *Fluid Dynamics Research*, **vol.12**(2), 61–93, August 1993.

[18] G.Russo and P.Smereka A remark on computing distance functions *Journal of Computational Physics*, **vol.163**(1), 51 – 67, 2000.

[19] J. A.Sethian and P.Smereka Level set methods for fluid interfaces *Annual Review of Fluid Mechanics*, **vol.35**(1), 341–372, 2003.

[20] C.-W.Shu *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, chapter Essentially Non-oscillatory and Weighted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws, pages 325–432 Springer Berlin / Heidelberg, 1998.

[21] C.-W.Shu and S.Osher Efficient implementation of essentially non-oscillatory shock-capturing schemes *Journal of Computational Physics*, **vol.77**(2), 439 – 471, 1988.

[22] P. D.Spelt A level-set approach for simulations of flows with multiple moving contact lines with hysteresis *Journal of Computational Physics*, **vol.207**(2), 389 – 404, 2005.

[23] M.Sussman, P.Smereka and S.Osher A level set approach for computing solutions to incompressible two-phase flow *Journal of Computational Physics*, **vol.114**(1), 146–159, September 1994.

[24] A. M.Worthington On the forms assumed by drops of liquids falling vertically on a horizontal plate *Proceedings of the Royal Society of London*, **vol.25**, 261–272, 1876.

[25] A.Yarin Drop impact dynamics: Splashing, spreading, receding, bouncing *Annual Review of Fluid Mechanics*, **vol.38**(1), 159–192, 2006.

Article [d]

Constrained reinitialisation of the conservative level set method

C. Walker and B. Müller

In J. Olsen and S. Johansen, editors, *Proceedings of the 8th International Conference on CFD in Oil & Gas, Metallurgical and Process Industries*, 2011

Erratum

- Equation (14) should read:

$$r^{i,j}_{(i,j)_\alpha} = \frac{\tilde{\psi}_{i,j}}{\tilde{\psi}_{(i,j)_\alpha}}$$

- The 5. step of the constrained reinitialisation should read:
  - If $M_{i,j} = M_{(i,j)_\alpha} = 1$, flag points for equal forcing
  - Else, set $r^{i,j}_{(i,j)_\alpha} = 0$, reduce $M_{i,j}$ by 1 and flag $\mathbf{x}_{(i,j)_\alpha}$ as point with $\beta_{(i,j)_\alpha} = 1$

CONSTRAINED REINITIALISATION OF THE CONSERVATIVE LEVEL SET METHOD

Claudio WALKER[1*], Bernhard MÜLLER[2†]

[1]NTNU Department of Energy and Process Engineering, 7491 Trondheim, NORWAY

[2]NTNU Department of Energy and Process Engineering, 7491 Trondheim, NORWAY

* E-mail: claudio.walker@ntnu.no
† E-mail: bernhard.muller@ntnu.no

ABSTRACT

In this paper it is shown that the reinitialisation of the conservative level function introduces spurious displacements of the interface. A method which prevents the interface in conventional level set methods from moving during the reinitialisation is adopted for the conservative level set method. It is shown that using the constrained reinitialisation with an adaptive distributed forcing retains the shape of the interface during reinitialisation. This eliminates the negative effects if too many reinitialisation steps are applied. However, our numerical experiments show that fixing of the interface during reinitialisation spoils the mass conservation of the conservative level set method.

Keywords: Multiphase flow, Level set method .

NOMENCLATURE

Greek Symbols

β	Forcing distribution factor.
Γ	Set of points addjacent to the interface.
δ	Least squares weights.
ε	Width parameter for the conservative level set function.
κ	Interface curvature.
ρ	Distance to domain center.
τ	Pseudo time.
ϕ	Conservative level set function.
Ψ	Stream function.
ψ	Signed distance function.

Latin Symbols

C	Set of points with nonzero foring term.
e	Error.
F	Forcing term.
f	Numerical flux.
h	Grid spacing.
$\mathbf{n}$	Interface normal.
m	Number of grid points.
t	Time.
S	Set of neighbouring points on opposit side of the interface.
$\mathbf{u}$	Velocity.
$\mathbf{x}$	Position.

Sub/superscripts

α	Index α.
i	Index i.
j	Index j.
n	Discrete time level.

INTRODUCTION

The level set method (LSM) is a popular method to describe the location of the interface in multiphase flow computations. It represents the interface with the help of a signed distance function which is advected by the fluid velocity. For a more thorough discussion of the LSM and its application to multiphase flow we refer the reader to the review by (Sethian and Smereka, 2003). This representation has the advantage of relatively simple calculations of interface normals and curvatures. Another often cited advantage of the LSM is that the parallelisation is straightforward. Due to the advection the level set function loses its signed distance property. Therefore it has to be reinitialised after a few advection steps. This is done by solving a reinitialisation equation. The deformation of the interface during this reinitialisation is a known problem and an explanation of the cause and a possible remedy was given by (Russo and Smereka, 2000). There exists a number of methods to reduce the displacement of the signed distance function during reinitialisation. Among these methods is the Constrained Reinitialisation (CR) (Hartmann *et al.*, 2008), which is minimising the displacement of the intersection points between the grid lines and the zero level set contour.

However, the LSM has an important disadvantage, it does not conserve the mass of the two fluids. Different approaches have been developed to satisfy the mass conservation of the level set method. Examples include the conservative level set method (CLSM) (Olsson and Kreiss, 2005) (Olsson *et al.*, 2007), the particle level set method (PLS) (Enright *et al.*, 2002) and the coupled level set/volume-of-fluid (CLSVOF) (Sussman and Puckett, 2000). The added complexity for both PLS and CLSVOF are significant. On the other hand the conservative level set method improves the mass conservation and keeps the simplicity of the original method.

We discovered that during the reinitialisation of the CLSM the interface is displaced considerably. In most applications of the CLSM this problem is not evident since typically only a few reinitialisation steps are conducted, and the deformation becomes only significant for high numbers of reinitialisation steps. There is no general rule on how frequent the CLSM has to be reinitialised and how many reinitialisation steps should be applied each time. It is therefore important to make sure that too many reinitialisation steps do not compromise the accuracy of the CLSM. Recently Hartmann et al. published a technique called high-order constrained reinitialisation (HCR) (Hartmann *et al.*, 2010) for the conventional LSM. But its implementation through a source term allows the adoption of HCR to do a constrained reinitialisation of the CLSM.

In the present paper a short introduction to the conservative level set method and its discretisation is given. Then we present an overview of the HCR and show how it can be adopted for the CLSM. Finally we show some numerical experiments and explain why it is after all not advisable to fix the interface during the reinitialisation of the CLSM.

CONSERVATIVE LEVEL SET METHOD

In level set methods the interface is defined as the iso contour of a smooth function. For ordinary level set methods this function is the signed distance from the interface, and the interface location is where the distance function is zero. The conservative level set function replaces the distance function by a hyperbolic tangent function ϕ with values between zero and one (Olsson and Kreiss, 2005). The position of the interface is located at the $\phi = 0.5$ contour line. Since we have smooth functions which are defined in the entire computational domain in both cases, we can easily extract additional geometrical information about the interface. For example the interface normal $\mathbf{n}$ and the curvature κ are defined as

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \tag{1}$$

and

$$\kappa = \nabla\cdot\mathbf{n}. \tag{2}$$

The interface is transported simply by advecting the level set function using the advection equation $\phi_t = -\mathbf{u}\cdot(\nabla\phi)$. If we have a divergence free velocity field, as it is the case for incompressible flow, the interface transport can be written as a conservation law.

$$\frac{\partial\phi}{\partial t} + \nabla\cdot(\mathbf{u}\phi) = 0 \tag{3}$$

Since all numerical methods will introduce an error as ϕ is advected, it will lose its hyperbolic tangent shape. The diffusion of the advection schemes will increase the distance in which ϕ rises from zero to one. (Olsson *et al.*, 2007) propose the following reinitialisation equation to force ϕ back to its hyperbolic tangent shape, which is solved to steady state with respect to the pseudo time τ:

$$\frac{\partial\phi}{\partial\tau} + \nabla\cdot\left(\phi(1-\phi)\hat{\mathbf{n}} - \varepsilon\left((\nabla\phi\cdot\hat{\mathbf{n}})\hat{\mathbf{n}}\right)\right) = 0, \tag{4}$$

where $\hat{\mathbf{n}}$ is the normal at the beginning of the reinitialisation, and ε determines the width of the hyperbolic tangent. It is important to note that also the reinitialisation equation is a conservation law. The first flux term causes a compression of the profile, whereas the second term is a diffusive flux. By multiplications with the normal $\hat{\mathbf{n}}$ there are only fluxes in the direction of the normal. This forced flux direction for both the compression and the diffusion term are essential to improve the mass conservation of the method. Indeed it was shown that the CLSM conserves mass as ε approaches 0. To illustrate the nature of Equation (4), we use a one-dimensional example. Suppose that the interface is located at $x = 0$. Then the normal reduces to $n = -1$ or $n = 1$. In the example we use the latter. In this case a steady state solution to Equation (4) is:

$$\phi = \frac{1}{2}\left(1 + \tanh\left(\frac{x}{2\varepsilon}\right)\right). \tag{5}$$

The solution is shown Figure 1, together with the compression and the diffusion terms. It is clearly visible that at steady state the compression and the diffusion are balanced. If ϕ would be too diffusive the compression term would outweigh the diffusive term and ϕ would be forced back to the steady state solution.

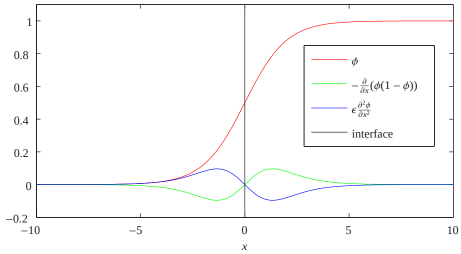

Figure 1: Illustration of the reinitialisation equation.

Discretisation

Advection

The divergence in the advection equation (3) is discretised by a 5th order finite difference weighted essentially non-oscillatory (WENO) scheme using a Lax-Friedrichs flux splitting (Shu, 1998). For time discretisation the 3rd order TVD Runge-Kutta method by (Shu and Osher, 1988) is applied.

Interface normal

In order to keep the stencils for the reinitialisation equation as small as possible (Desjardins *et al.*, 2008) proposed to compute the fluxes of the reinitialisation equation (4) at the cell faces. As a consequence the normal is required at the cell faces. It was also proposed by the authors to compute the normal from a signed distance function ψ which in turn is computed using a fast marching method (FMM). The reason for this is that the gradient of the conservative level set function ϕ becomes extremely small far away from the interface. In this region with small gradients small errors in ϕ can lead to a normal pointing in the wrong direction. Close to the interface, however, the gradient of ϕ is large enough such that a reliable normals can be computed directly form ϕ. Further away from the interface the accuracy of the normal is not so important anymore, but they should be continuous and not contain spurious oscillations. In the present work the normal are computed directly form the conservative level set function where $0.0001 \leq \phi \leq 0.9999$, otherwise they are computed from a signed distance function.

The gradient for the normal on the cell faces in x-direction at $x = x_{i-1/2,j}$ is approximated by the following stencil:

$$(\nabla\phi)_{x\,i-1/2,j} \approx \frac{\phi_{i,j} - \phi_{i-1,j}}{h} \tag{6}$$

$$(\nabla\phi)_{y\,i-1/2,j} \approx \frac{\phi_{i-1,j+1} + \phi_{i,j+1} - \phi_{i-1,j-1} - \phi_{i,j-1}}{4h}. \tag{7}$$

The gradient at the cell faces in y-direction $(\nabla\phi)_{i,j-1/2}$ is treated analogously. Finally the normal can be computed from the gradient using equation (1). Note grid points where $|\nabla\phi| = 0$ have to be treated separately, here we chose to fill the normal at these points with a unit vector pointing in an arbitrary direction.

Reinitialisation

A central difference method on a staggered grid is used to discretise the reinitialisation equation (4). The fluxes in x-direction as well as those in y-direction are computed on the corresponding cell faces. First the convective fluxes are obtained at the cell centers from the level set function i.e. $f_{c\,i,j} = \phi_{i,j}(1-\phi_{i,j})$ and then they are interpolated to the cell faces using linear interpolation e.g. $f_{c\,i+1/2,j} = \left(f_{c\,i,j} + f_{c\,i+1,j}\right)/2$. The diffusive fluxes are computed directly at the cell faces to keep the stencil as small as possible. Accordingly the diffusive fluxes at the cell faces in x-direction read:

$$f_{d\,i-1/2,j} = \varepsilon\left((\nabla\phi)_{x\,i-1/2,j}\cdot n_{x\,i-1/2,j} + (\nabla\phi)_{y\,i-1/2,j}\cdot n_{y\,i-1/2,j}\right), \tag{8}$$

where $n_{x\,i-1/2,j}$ and $n_{y\,i-1/2,j}$ are the x- and y-components of the interface normal. This leads to the total fluxes in x- and y-directions:

$$f_{i-1/2,j} = \left(f_{c\,i-1/2,j} - f_{d\,i-1/2,j}\right)\cdot n_{x\,i-1/2,j} \tag{9}$$

$$g_{i,j-1/2} = \left(f_{c\,i,j-1/2} - f_{d\,i,j-1/2}\right)\cdot n_{y\,i,j-1/2}\,. \tag{10}$$

Eventually the total residual of the reinitialisation equation is:

$$\frac{\partial\phi}{\partial\tau} \approx -\frac{f_{i+1/2,j} - f_{i-1/2,j}}{h} - \frac{g_{i,j+1/2} - g_{i,j-1/2}}{h}. \tag{11}$$

The forward Euler scheme is used to advance the conservative level set function in the pseudo time τ. To ensure stability the time step must fulfill:

$$\Delta\tau\left(\frac{2}{h} + \frac{4}{h^2}\varepsilon\right) \leq 1. \tag{12}$$

CONSTRAINED REINITIALISATION

We observed that the convergence of the reinitialisation with the described discretisation was poor for simple cases. In Figure 2 the result of the reinitialisation of a circle is shown. During the first few iterations the residual drops fast and then the convergence is slowed down. During this period of small convergence rate the interface develops towards a rhombus. This tendency to deform the interface is not reduced as the grid is refined. In most applications this defect is not evident since typically only a few reinitialisation steps are conducted and the residual is not reduced until the numerical steady state.

Constrained reinitialisation for conventional level set methods

A similar problem appears in conventional level set methods. Recently (Hartmann *et al.*, 2010) proposed a method to reduce the displacement of the signed distance function ψ during reinitialisation. The idea is to add a source term to the residual of the differential equation, which minimises the displacement of the intersection points between the zero contour of ψ and the grid lines in a least squares sense. If two grid points $\mathbf{x}_{i-1,j}$ and $\mathbf{x}_{i,j}$ which are located on opposite sides of the interface, the condition that the linear interpolation of the intersection point between the zero contour line and the grid line between those two points does not move during reinitialisation, can be reduced to $\frac{\tilde\psi_{i,j}}{\tilde\psi_{i-1,j}} = \frac{\psi_{i,j}}{\psi_{i-1,j}}$, where $\tilde\psi$ and ψ are the signed distance functions before and after reinitialisation, respectively. In general a grid point can have several neighbours which are on the opposite side of an interface. The previous condition cannot be fulfilled for all involved neighbours, since the problem is overdetermined. Let $S_{i,j}$ be the set of all neighbouring grid points of $\mathbf{x}_{i,j}$ which are on the opposite side of the interface, and $M_{i,j}$ the number of grid points in $S_{i,j}$. Further we denote an arbitrary point in $S_{i,j}$ by $\mathbf{x}_{(i,j)\alpha}$ such that $\alpha = 1 \ldots M_{i,j}$. In the constrained reinitialisation CR-1 (Hartmann *et al.*, 2010) the following least squares function is minimised.

$$L_{i,j} = \sum_{\alpha=1}^{M_{i,j}} \delta^{\alpha}_{i,j}\left(\psi_{i,j} - \psi_{(i,j)\alpha}\cdot r^{i,j}_{(i,j)\alpha}\right)^2, \tag{13}$$

$$r^{i,j}_{(i,j)\alpha} = \frac{\tilde\psi_{i,j}}{\tilde\psi_{i-1,j}} \tag{14}$$

(Hartmann *et al.*, 2010) chose the weights $\delta^{\alpha}_{i,j} = 1$. If $L_{i,j}$ is derived with respect to $\psi_{i,j}$ and this is set equal to zero we get the target value for the distorted signed distance function, such that it minimises the displacement of the interface.

$$T_{i,j} = \frac{\sum_{\alpha=1}^{M_{i,j}} \delta^{\alpha}_{i,j}\left(\psi_{(i,j)\alpha}\cdot r^{i,j}_{(i,j)\alpha}\right)}{\sum_{\alpha=1}^{M_{i,j}} \delta^{\alpha}_{i,j}} \tag{15}$$

Finally the CR-1 correction term at the nth reinitialisation time step is formulated as the difference between $T^n_{i,j}$ and $\psi^n_{i,j}$:

$$F^n_{i,j} = \frac{\beta_{i,j}}{h}\left(\psi^n_{i,j} - T^n_{i,j}\right). \tag{16}$$

Here $\beta_{i,j}$ is a coefficient which distributes the correction between neighbouring grid points. For consistency the sum $\beta_{i,j} + \beta_{(i,j)\alpha}$ must always be equal to 1. Setting $\beta_{i,j} = 0.5$ corresponds to do half of the correction on either side of the interface. Since the forcing leads to an instability when one of points of the set $S_{i,j}$ changes its sign during reinitialisation,

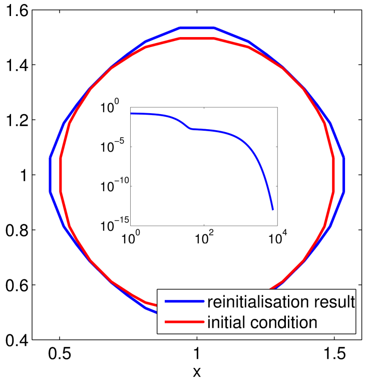

Figure 2: Reinitialisation result of a circle with 8 grid points per diameter, the small insert shows the evolution of the reinitialisation residual over the number of time levels.

the forcing is not applied at the grid points in this particular $S_{i,j}$.

Adaptation to the conservative level set method

The idea is to apply the CR-1 forcing to the reinitialisation of the CLSM to avert an undesirable movement of the interface during reinitialisation. In the CLSM the interface is given as the 0.5 contour line. Therefore ψ in equation (13) to (16) must be replaced by $\phi^- = \phi - 0.5$.

Stabilising CR-1

An essential difference between the signed distance function ϕ and the conservative level set function ψ is that the latter has larger gradients at the interface. As a consequence the forcing term for the CLSM tends to be stronger. For points where the distance to the interface approaches h the magnitude of the forcing term is increasing rapidly. It was observed in numerical experiments that if the distance between the intersection point and the grid point becomes smaller than $0.01h$ the constrained reinitialisation of ϕ becomes unstable. Imagine two bars which cross each other. Suppose one is only allowed to move one end of one bar to adjust the position of the crossing point. If now the crossing point is close to the fixed end of the moving bar, the free end has to be moved relatively far to obtain a given displacement of the crossing point. On the other hand if one could move the end which is closer to the crossing point a much smaller movement is required to obtain the same displacement of the crossing point. The correction can also be divided to both ends where the necessary corrections at each end are multiplied by factors which have to add to one. These factors are the distribution factors $\beta_{i,j}$.

In order to stabilise the constrained reinitialisation of the CLSM the forcing for pairs where the crossing point is close to one of the points should be distributed better, which means adjusting $\beta_{i,j}$ in equation (16). For pairs where both neighbours have $M_{i,j} = 1$, $\beta_{i,j}$ can be chosen such that the forcing term has the same magnitude on both sides. As an example if both $M_{i,j}$ and $M_{i-1,j}$ are equal to 1 and the crossing point is either close to $\mathbf{x}_{i,j}$ or $\mathbf{x}_{i-1,j}$ set

$$\beta_{i,j} = \left(\frac{\psi_{i,j} - T_{i,j}}{\psi_{i-1,j} - T_{i-1,j}} + 1 \right) \tag{17}$$

$$\beta_{i-1,j} = 1 - \beta_{i,j}. \tag{18}$$

This choice of distribution coefficients ensures that the forcing term is equal for both points.

In general $M_{i,j} > 1$ especially for points which are close to the interface. There $\beta_{i,j}$ cannot be adjusted to an arbitrary value since $\beta_{i,j} + \beta_{(i,j)_\alpha} = 1$, and the set $S_{i,j}$ often overlaps with the sets from the neighbouring points. Therefore setting $\beta_{i,j}$ gives a condition to a number of β from neighbouring grid points. A solution to keep the effect of choosing $\beta_{i,j}$ local is to use the weights $\delta^{\alpha}_{i,j}$ in the least squares function (13). If a grid point $\mathbf{x}_{i,j}$ is too close to an intersection point $\beta_{i,j}$ is set to 1. This would require to set all $\beta_{(i,j)_\alpha} = 0$. Instead we choose $\delta^{\hat{\alpha}}_{(i,j)_\alpha} = 0$ in all the neighbouring cells where $\hat{\alpha}$ points to cell (i,j). That is equivalent to do the complete forcing on the grid point close to the intersection and ignoring the constraint at all the neighbouring points.

Summary of the constrained reinitialisation

Only at points adjacent to the interface the operations for the level set forcing have to be conducted, we define therefore the set

$$\Gamma = \left\{ \mathbf{x}_{i,j} \colon \left(\phi^-_{i,j} \phi^-_{i',j} \leq 0 \right) \vee \left(\phi^-_{i,j} \phi^-_{i,j'} \leq 0 \right) \right\}, \tag{19}$$

where $i' \in \{i-1, i+1\}$ and $j' \in \{j-1, j+1\}$. For sets $S_{i,j}$ where ϕ at one of the points changes its sign during reinitialisation the forcing is not applied. To that effect the following set of grid points will have a forcing term $F_{i,j} \neq 0$:

$$C^n = \left\{ \mathbf{x}_{i,j} \in \Gamma \colon \phi^{-n}_{i,j} \phi^{-n}_{(i,j)_\alpha} < 0 \, \forall \alpha \in \{1, ..., M_{i,j}\} \right\}. \tag{20}$$

Using these definitions the steps for the constrained reinitialisation of the CLSM are:

1. Compute the interface normal. and (7))
2. Compute the shifted conservative level set function before reinitialisation $\tilde{\phi}^- = \tilde{\phi} - 0.5$.
3. Find all points in Γ.
4. Update the set $S_{i,j}$, and compute $r^{i,j}_{(i,j)_\alpha}$ for all points in Γ.
5. Find points where $r^{i,j}_{(i,j)_\alpha} <$ TOL
   - If $M_{i,j} = M_{(i,j)_\alpha} = 1$, flag points for equal forcing
   - Else, set $r^{i,j}_{(i,j)_\alpha} = 0$, reduce $M_{i,j}$ by 1 and flag $\mathbf{x}_{i,j}$ as point with $\beta_{i,j} = 1$
6. For all points with $\beta_{i,j} = 1$, do for all (i', j') where $\mathbf{x}_{(i',j')_{\hat{\alpha}}} = \mathbf{x}_{i,j}$, $\delta^{\hat{\alpha}}_{i',j'} = 0$ and reduce $M_{i',j'}$ by 1 where $\mathbf{x}_{(i',j')_{\hat{\alpha}}} = \mathbf{x}_{i,j}$.
7. Remove points $\mathbf{x}_{i,j}$ where $M_{i,j} = 0$ from Γ.
8. Solve the constrained reinitialisation problem by performing the following steps for each iteration:
   (a) Compute the shifted conservative level set function at pseudo time step n: $\phi^{-n} = \phi^n - \frac{1}{2}$.
   (b) Update the set C^n
   (c) For all grid points in C^n compute the forcing terms given by equation (16), use $\beta_{i,j} = 0.5$ unless the point is flagged for $\beta_{i,j} = 1$ or equal correction.
   (d) Compute residual from the PDE (4).
   (e) Advance ϕ one pseudo time step.

NUMERICAL EXPERIMENTS

Reinitialisation

To test the constrained reinitialisation method described in the previous section we consider a circle with radius $R = \frac{1}{2}$ which is placed in the center of a square domain with unit length. The initial condition is $\tilde{\phi} = \left(1 + \exp\left(\frac{3(\rho - R)}{2\varepsilon}\right)\right)^{-1}$, where $\rho = \sqrt{\left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2}$ is the distance from the center of the domain. For this case the exact steady state solution of equation (4) is:

$$\phi_{\text{exact}} = \left(1 + \exp\left(\frac{\rho - R}{\varepsilon}\right)\right)^{-1}. \tag{21}$$

The reinitialisation equation was iterated until the 2-norm of the residual was below 10^{-13}. The error norm is computed as follows:

$$e = \frac{1}{m}\left(\sum_{i,j=1}^{m}\left(\phi_{\text{exact}\,i,j}-\phi_{i,j}\right)^2\right)^{\frac{1}{2}}, \tag{22}$$

where m is the number of grid points in each direction. To assess the deformation of the interface the reinitialised ϕ was interpolated to a fine grid with 500 points in each direction using bi-cubic splines. On this fine grid the position of all intersection points between the grid lines and the 0.5 contour line were computed using a 1-dimensional cubic spline interpolation. The position error of the interface e_p is then given as the 2-norm of the distance between each interpolated intersection point and its correct position.

Analytic normal

The errors after the reinitialisation with analytic normal and a fixed $\varepsilon = 0.044$ are presented in Figure 3. The error norm e is reduced with second order which is consistent with the discretisation scheme. To find the position of the interface we need to interpolate it. This interpolation is the reason that the position error of the interface is one order higher than the order of the discretisation scheme. Special attention should be payed to the results with $m = 81$. At this grid resolution 16 grid points were marked since their distance to the interface was smaller than $0.01h$. These grid points where treated with the procedure explained previously. Without this improved distribution of the forcing term the reinitialisation is unstable for $m = 81$. As it can be seen in Figure 3, the convergence rate of the position error is slightly reduced due to the unequal distribution of the forcing term. For comparison the errors without the CR-1 are plotted in Figure 3 as well. Since the interface is developing a rhombus shape (see Figure 2) at all grid resolutions neither the error norm nor the position error are converging with grid refinement. In Table 1 we list the number of reinitialisation steps which were required to reduce the residual to the target value of 10^{-13}. The constrained reinitialisation accelerates the convergence of the residual significantly. For the unconstrained reinitialisation of the circle with $m = 64$ and $m = 81$ the target residual was not reached after 100000 reinitialisation steps. The residuals at $n = 100000$ were $1.55 \cdot 10^{-5}$ and $4.03 \cdot 10^{-5}$ for $m = 64$ and $m = 81$, respectively.

Table 1: Number of iterations required to reach a residual of 10^{-13} during the reinitialisation of a circle.

m	8	16	32	64	81
CR-1	165	318	943	3654	4760
without	372	7536	96637	> 100000	> 100000

Numeric normal

The same test as in the previous section is repeated, but the interface normal is computed numerically. For a constant $\varepsilon = 0.044$ the results are almost identical to the reinitialisation with the analytic interface normal. Since the mass conservation of the CLSM is improved as ε is decreased, the reinitialisation of the circle was repeated once with $\varepsilon = 0.2\sqrt{h}$ and with $\varepsilon = 0.6h$. As ε is decreased the width of tangent hyperbolic is also decreased, which means that there are less grid points resolving the area where ϕ changes from 0 to 1. The result is that the convergence rate is reduced if ε is decreasing together with the grid width. The errors for simultaneous reduction of h and ε are shown in Figure 4.

Advection and Reinitialisation

Initially a circle of radius $R = 0.1$ is placed at $(0.5, 0.7)$ in a square domain with unit length. The circle is advected with an external velocity field with the stream function

$$\Psi = -\pi\left(x^2 + y^2 - x - y\right). \tag{23}$$

This velocity field is advecting the circle anticlockwise around the center of the computational domain. To keep the error contribution from the temporal discretisation negligible a small CFL number of 0.05 was chosen for all examples. After 100 time steps a reinitialisation was performed. During the reinitialisation the residual was reduced to 10^{-8} or a maximum of 500 iterations were done. The convergence of the errors for the constrained CLSM is plotted in Figure 5 for

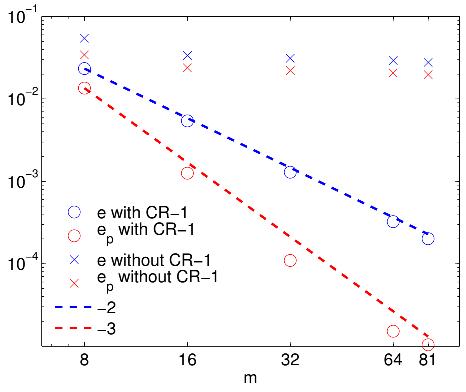

Figure 3: Error convergence for the reinitialisation of a circle with analytic interface normal and $\varepsilon = 0.044$.

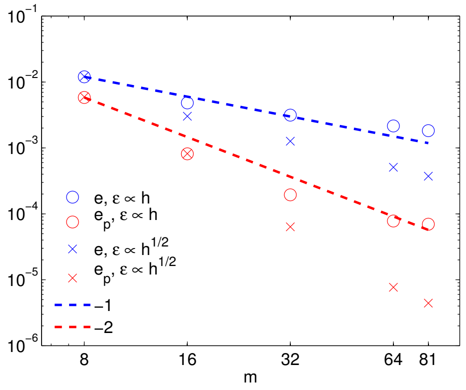

Figure 4: Error convergence for the reinitialisation of a circle with numeric interface normal and $\varepsilon \propto h$ or $\varepsilon \propto \sqrt{h}$.

$\varepsilon = 0.1 \cdot m^{-1/2}$. Figure 6 shows the interface position after one revolution for $m = 51$. For the case where the reinitialisation was done with CR-1 the interface retains a circular shape but the area is reduced significantly. On the other hand, if the unconstrained reinitialisation is used, the area is conserved but the interface loses its circular shape. Here we exaggerated this effect by the large number of reinitialisation steps. If e.g. every 100 advection steps only 10 reinitialisation steps are conducted, the interface position error is reduced by almost a factor of 16. If the interface is only advected and never reinitialised, the area loss is smaller than for the CR-1 reinitialisation and the circular shape is retained. The area loss for the three different reinitialisations is summarised in Table 2. The reduction of ε leads only for the CLSM without CR-1 to a reduction in the area change. For the constrained CLSM and if the circle is only advected, the area change decreases with increasing ε. This is due to the fact that the conservative level set function is less steep for larger ε, and therefore the diffusive error of the advection scheme is reduced.

The large mass loss for the constrained reinitialisation can be explained as follows. Initially the level set function can be compared with a cylinder with height 1 and the radius of the circle. If the circle has a radius of 1 then the volume of the cylinder is π. Since the numerical advection is diffusive, we assume the advected cylinder will transform to the shape of a cone with height of 1. The conservative advection will conserve the volume of the cylinder. For this reason the radius of the cone at a height of 0.5 will be only $\frac{\sqrt{3}}{2}$ instead of 1. During a constrained reinitialisation a new cylinder with this smaller cross section will be created and the cycle starts again.

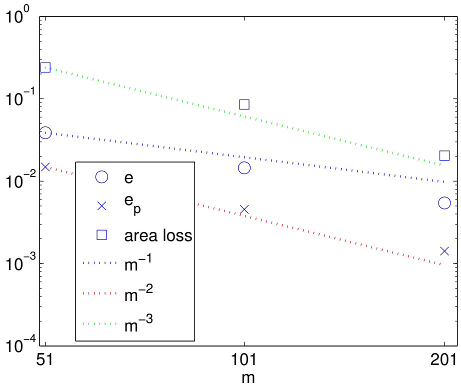

Figure 5: Errors for the advected circle after one revolution.

Table 2: Area change in % for the advected circle with $m = 51$.

ε	0.1	0.2	0.3
CLSM CR-1	−23.98	−11.14	−4.93
CLSM	0.10	0.60	0.65
Advection	14.49	5.60	1.43

CONCLUSION

We applied the constrained reinitialisation CR-1 by (Hartmann *et al.*, 2010) to the reinitialisation of the conservative level set function. The larger modulus of the gradient of the conservative level set function can cause instabilities during the constrained reinitialisation, if the interface is located close to a grid point. This instability can be avoided, if the forcing is not divided equally between two neighbouring points. Evidence was given that the constrained reinitialisation prevents spurious interface deformation during the reinitialisation independent of the number of reinitialisation steps, and that the convergence of the residual is accelerated. However, numerical experiments showed that preventing the interface from moving during reinitialisation spoils the mass conservation of the CLSM. It was also shown that care has to be taken that the number of reinitialisation steps of the unconstrained CLSM is not too large such that the reinitialisation introduces spurious interface deformations.

REFERENCES

DESJARDINS, O. *et al.* (2008). “An accurate conservative level set/ghost fluid method for simulating turbulent atomization”. *Journal of Computational Physics*, **227(18)**, 8395–8416.

ENRIGHT, D. *et al.* (2002). “A hybrid particle level set method for improved interface capturing”. *Journal of Computational Physics*, **183(1)**, 83–116.

HARTMANN, D. *et al.* (2008). “Differential equation based constrained reinitialization for level set methods”. *Journal of Computational Physics*, **227(14)**, 6821 – 6845.

HARTMANN, D. *et al.* (2010). “The constrained reinitialization equation for level set methods”. *Journal of Computational Physics*, **229(5)**, 1514 – 1535.

OLSSON, E. and KREISS, G. (2005). “A conservative level set method for two phase flow”. *Journal of Computational Physics*, **210(1)**, 225–246.

OLSSON, E. *et al.* (2007). “A conservative level set method for two phase flow II”. *Journal of Computational Physics*, **225(1)**, 785–807.

RUSSO, G. and SMEREKA, P. (2000). “A remark on computing distance functions”. *Journal of Computational Physics*, **163(1)**, 51 – 67.

SETHIAN, J.A. and SMEREKA, P. (2003). “Level set

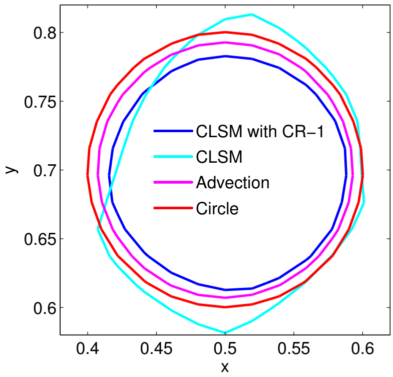

Figure 6: Advected circle, interface position after one revolution.

methods for fluid interfaces". *Annual Review of Fluid Mechanics*, **35(1)**, 341–372.

SHU, C.W. (1998). *Essentially Non-oscillatory and Weighted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws*, in: A. Quarteroni (ed.), Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, 325–432. Springer Berlin / Heidelberg.

SHU, C.W. and OSHER, S. (1988). "Efficient implementation of essentially non-oscillatory shock-capturing schemes". *Journal of Computational Physics*, **77(2)**, 439 – 471.

SUSSMAN, M. and PUCKETT, E.G. (2000). "A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows". *Journal of Computational Physics*, **162(2)**, 301–337.

Article [e]

Microscale enhancement of macroscale modeling for capillary-driven contact line dynamics

M. Kronbichler, C. Walker, G. Kreiss, and B. Müller

Submitted, 2012

Article [f]

Contact line treatment with the sharp interface method

C. Walker and B. Müller

Submitted, 2012

Is not included due to copyright