Mass Conserving Simulations of Two Phase Flow

ELIN OLSSON

Licentiate Thesis
Stockholm, Sweden 2006

© Elin Olsson, 2006

Tryck: Universitetsservice US AB
Consider a mixture of two immiscible, incompressible fluids e.g. oil and water. Since the fluids do not mix, an interface between the two fluids will form and move in time. The motion of the two fluids can be modelled by the incompressible Navier-Stokes equations for two phase flow with surface tension together with a representation of the moving interface. The parameters in the Navier-Stokes equations will depend on the position and other properties of the interface. The interface should move with the velocity of the flow at the interface. Since the fluids are incompressible, the density of each fluid is constant. Mass conservation then implies that the volume occupied by each of the two fluids should not change with time. The object of this thesis has been to develop a new numerical method to simulate incompressible two phase flow accurately that conserves mass and volume of each fluid correctly.

Numerical simulations of incompressible two phase flow with surface tension have been a challenge for many years. Several methods have been developed and used prior to the work presented in this thesis. The two most commonly used methods are volume of fluid methods and level set methods. There are advantages and disadvantages of both of the methods.

In volume of fluid methods the interface is represented by a discontinuity of a globally defined function. Because of the discontinuity it is hard both to move the interface as well as to calculate properties of the interface such as curvature. Specially designed methods have to be used, and all these methods are low order accurate. Volume of fluid methods do however conserve the volumes of the two fluids correctly.

In level set methods the interface is represented by the zero contour of the globally defined signed distance function. This function is smooth across the interface. Since the function is smooth, standard methods for partial differential equations can be used to advect the interface accurately. A reinitialization is however needed to make sure that the level set function remains a signed distance function. During this process the zero contour might move slightly. Because of this, the volume conservation of the method becomes poor.

In this thesis we present a new level set method. The method is designed such that the volume of each fluid is conserved, at least approximately. The interface is represented by the 0.5 contour of a regularized characteristic function. As for standard level set methods, the interface is moved first by an advective step, and then reinitialized. Unlike traditional level set methods, we can formulate the reinitialization as a conservation law. Conservative methods can then be used to move and to reinitialize the level set function numerically. Since the level set function is a regularized characteristic function, we can expect good conservation of the volume bounded by the interface.

The method is discretized using both finite differences and finite elements. Uniform and adaptive grids are used in both two and three space dimensions. Good convergence as well as volume conservation is observed. Theoretical studies are performed to investigate the conservation and the computational time needed for reinitialization.
## Contents

1 Incompressible Two Phase Flow  
  1.1 Mathematical Formulation ........................................... 5  
  1.2 Traditional Numerical Methods for Two Phase Flow ............... 6  
  1.3 The Volume of Fluid Method ......................................... 6  
  1.4 The Level Set Method .................................................. 7  
  1.5 Front Tracking Methods ............................................... 8  
  1.6 Numerical Treatment of Discontinuities ............................. 9

2 A Conservative Level Set Method  
  2.1 Introduction ............................................................. 11  
  2.2 Summary of Included Papers ......................................... 11

3 Open Questions and Future Work  
  3.1 Contact Lines .......................................................... 15  
  3.2 Analysis of the Method ............................................... 15  
  3.3 Applications in Petroleum Engineering and Material Science .... 16

Bibliography  

v
Preface

This thesis consists of an introduction and two papers. In both papers the ideas were developed by the two authors together. The author of this thesis also wrote both manuscripts and made all the numerical calculations.


**Paper 2** E. Olsson, G. Kreiss. A conservative level set method for two phase flow II. Submitted.
Acknowledgments

First of all I want to thank my supervisor professor Gunilla Kreiss for her devotion, her skills and her ability to educate and to cooperate.

I also want to thank all my colleagues at NADA and the members of the Computational Phase Transformation project for helpful discussions and valuable friendship.

Financial support from SSF, Knut and Alice Wallenbergs stiftelse, NORFA/NordForsk and Ragnar and Astrid Signeuls stiftelse is gratefully acknowledged.

Tack mamma och pappa, Frida, Magnus och Alba, och Cecilia för allt stöd och all vänskap.

Sist men inte minst, tack Claude för att du alltid stöttar, uppmuntrar och tror på mig.
Chapter 1

Incompressible Two Phase Flow

Many physical phenomena can be modeled as incompressible, immiscible two phase flow with surface tension. Examples include water droplets falling through the air, petroleum and water mixtures flowing in a porous media, molten glass in air etc. Since the fluids are immiscible, a moving interface between the fluids will form. In this thesis we will present a new method to simulate the motion of the two fluids and the interface between them. To justify the need for the new method, we will give a short overview of the most frequently used traditional techniques. As we will see, the main advantages of the new method is that it conserves mass correctly and that it is easy to implement for problems in both two and three dimensions.

1.1 Mathematical Formulation

Assume a given domain $\Omega$ occupied by two different fluids. Denote the subdomains occupied by fluid 1 and 2 by $\Omega_1$ and $\Omega_2$ respectively. We assume that $\Omega \setminus \Omega_1 = \Omega_2$. The internal boundary between $\Omega_1$ and $\Omega_2$ we denote by $\Gamma$.

Taking forces due to inertia, viscosity, gravitation and surface tension into account, conservation of momentum gives

$$\left(\rho \vec{u}\right)_t + \nabla \cdot (\rho \vec{u} \vec{u}) = -\nabla p + \frac{1}{Re} \nabla \cdot \left(\mu \left(\nabla \vec{u} + (\nabla \vec{u})^T\right)\right) + \frac{\rho}{Fr^2} \vec{e}_g + \frac{1}{We} \vec{F}_{st}. \tag{1.1}$$

If the fluids are incompressible, conservation of mass gives

$$\nabla \cdot \vec{u} = 0. \tag{1.2}$$

The system (1.1) and (1.2) is known as the incompressible Navier-Stokes equations. $Re$, $Fr$ and $We$ are non dimensional parameters depending on properties of the fluids and the length scale of the problem. $\rho$ and $\mu$ are dimensionless density and viscosity and $\vec{F}_{st}$ is the force due to surface tension. $\vec{u}$ and $p$ are the unknown velocity and pressure. In addition to the Navier-Stokes equations, the motion of the interface has to be described. Given an initial interface, the interface should move with the velocity of the flow at the interface. This can be formulated as

$$\frac{dX}{dt} = \vec{u}(X), \quad X \in \Gamma. \tag{1.3}$$

(1.3) gives an ordinary differential equation (ODE) for every point on the interface. It is possible to represent the boundary by a certain level set of a globally defined function $\Phi$. 

5
The ODEs (1.3) can then be replaced by the partial differential equation (PDE)

$$\Phi_t + \vec{u} \cdot \nabla \Phi = 0$$

on all of $\Omega$.

We will assume that the density and viscosity is constant within each of the fluids, i.e.

$$\rho(x) = \begin{cases} 
\rho_1 & \text{if } x \in \Omega_1 \\
\rho_2 & \text{if } x \in \Omega_2 
\end{cases}$$

and respectively for $\mu$. The force due to surface tension can be written as

$$\vec{F}_{st}(x) = \kappa(x)\hat{n}(x)\delta_{\Gamma}.$$ 

Here, $\kappa$ and $\hat{n}$ are the mean curvature and normal of the interface. $\delta_{\Gamma}$ is a Dirac function with support only on $\Gamma$ such that $\int_{\Omega} \delta_{\Gamma} d\Omega = \int_{\Gamma} ds$. In order to include surface tension the normal and curvature of the interface are thus needed.

The density and viscosity will in general be discontinuous across the interface. These discontinuities together with the surface tension will result in discontinuities in pressure and gradients of the velocity.

It is important to notice that the volume occupied by each fluid will be conserved in time. This means that if there is no mass flow through the boundaries of $\Omega$ then $\frac{\partial}{\partial t} \left( \int_{\Omega_1} d\Omega \right) = 0$. This is a consequence of the incompressibility and the conservation of mass.

### 1.2 Traditional Numerical Methods for Two Phase Flow

In general only approximate solutions of the equations (1.1), (1.2) and (1.3) can be found. Several different techniques have been developed for incompressible two phase flow. One main difference between the methods is how the interface is represented. How the curvature and normal are calculated will naturally depend on the representation.

All methods for representing the interfaces can be divided into two classes. In the first class, the interface $\Gamma$ is represented implicitly by a function defined on all of $\Omega$. To this class belong the level set method, the volume of fluid method and the phase field method. In the second class, the interface $\Gamma$ is instead explicitly tracked. These methods are called front tracking methods.

Among these methods, the volume of fluid method, the level set method and front tracking methods are probably the most commonly used. A short summary of the basic features of these methods will be given in the following sections. For phase field methods we refer the review paper by Anderson, McFadden and Wheeler [1] and to the work of Jacqmin [7].

### 1.3 The Volume of Fluid Method

The volume of fluid method has since it was first introduced in 1976 by Noh and Woodward [9] been extensively used, studied and developed. Here, we will only describe the main idea of the method. A more complete overview of the method and its developments can be found in the review paper by Scardovelli and Zaleski [14].

The idea of the volume of fluid method is the following. The computational domain $\Omega$ is divided into small cells. Each cell is assigned a value between zero and one representing
the fraction of one of the fluids within that cell. From these cell fractions the position of
\( \Gamma \) can be determined using some kind of reconstruction. In the original paper by Noh and
Woodward [9], a piecewise constant reconstruction was used. Later, many piecewise linear
reconstructions have been proposed, e.g. by Rider and Kothe [13] and Pilliod and Puckett [12]. A characteristic function \( \chi \) being zero within fluid one and one within fluid two can
now be determined.

To advect the interface, the PDE
\[
\chi_t + \nabla \cdot (\vec{u}\chi) = 0
\]
can be solved. This is done using finite volume techniques, i.e. the flux of each fluid in and
out of each cell is calculated. Because of the discontinuity in \( \chi \), special methods have to be
used not to smear the discontinuity. The order of accuracy seems to range from one to two
for volume of fluid methods. The method does however conserve mass exactly. This is not
the case for the level set method and front tracking methods.

To calculate normals and curvature, a smooth version of \( \chi \), \( \tilde{\chi} \), is in general calculated
using some kind of regularization. The normal and curvature can then be computed from
\( \hat{n} = \nabla \tilde{\chi} / |\nabla \tilde{\chi}| \) and then \( \kappa = -\nabla \cdot \hat{n} \).

1.4 The Level Set Method

The idea to evolve an interface by representing it as a level set of a global function was
introduced by Osher and Sethian [11]. They described techniques for motion in either a
given underlying velocity field or with curvature dependent speeds. Since then, the level
set method have been applied to several different problems such as propagation of fronts,
crystal growth, image processing, compressible and incompressible flow etc.

In level set methods, the interface is represented as the zer o contour of the signed distance
function \( \Phi \):
\[
\Phi(x) = \begin{cases} 
\min_{x_I \in \Gamma} \|x - x_I\| & \text{if } x \in \Omega_1 \\
-\min_{x_I \in \Gamma} \|x - x_I\| & \text{if } x \in \Omega_2.
\end{cases}
\]
The zero contour, and thus the interface, can now be moved correctly by solving the PDE
\[
\Phi_t + \vec{u} \cdot \nabla \Phi = 0.
\] (1.4)

In contrast to the characteristic function used in the volume of fluid method, this function
is smooth across the interface \( \Gamma \). Because of this, standard higher order techniques can
be used to solve (1.4) numerically. Mostly, higher order essentially non-oscillatory (ENO)
schemes ([6], [16], [17]) are used.

(1.4) will move the zero contour in the right way. \( \Phi \) will however in general not remain
a signed distance function as time evolves. A reinitialization is therefore needed to make
sure that \( \Phi \) remains a distance function. This reinitialization can also be formulated as a
PDE. Given an initial \( \Phi_0 \) with the correct position of the zero contour,
\[
\Phi_t + \text{sgn}(\Phi_0) (|\nabla \Phi| - 1) = 0
\] (1.5)
can be solved to steady state. This was used by Sussman, Smereka and Osher [21], where
the level set method was first used for incompressible two phase flow calculations. For
numerical reasons, a regularized sign function was used. In numerical approximations of
(1.5), it has become evident that the zero contour will move, in particular if the initial \( \Phi_0 \)
is far from a signed distance function. This in turn implies a poorer conservation of the
volume bounded by the interface. Several different approaches have been used in order to
decrease the movement of the zero contour during reinitialization. A modification to the reinitialization was presented by Sussman and Fatemi [18] and also used in [19]. There, a constraint was added to the reinitialization such that the volume bounded by the zero contour could be approximately conserved locally within each cell. The zero contour might thus still move, but the volume bounded by the zero contour will be approximately preserved during the reinitialization. It should be noted that the numerical solution of the advection will also imply a small change in the volume bounded by the interface. Even though the results in [19] shows better conservation compared to [21], there is still a continuous mass loss which accumulates as time evolves.

Other approaches have been used to improve the mass conservation of level set methods. Enright et al [4] proposed to distribute marker particles close to the zero level set. These marker particles were also moved with the flow, but by solving the ODE (1.3) for each particle. Numerical experiments have shown that methods based on the ODE formulation have better conservation properties compared to the level set method. Any marker particle initially on one side of the interface should always remain on that side. Marker particles moving from one side to the other side of the zero contour of the level set function was used to indicate where the interface had moved incorrectly. Within each cell closer than $3\Delta x$ to the interface, 16 for 2D calculations and 64 for 3D calculations marker particles were placed. These marker particle will improve the resolution close to the zero contour. In particular, the particle level set method improves resolution where features of the interface cannot be resolved by the fixed grid used for the level set function. Numerical evidence of improved resolution and conservation during both advection and reinitialization was presented. No guarantee of conservation for long time simulation can however be given. The marker particles obviously increases the complexity of the level set method. The method has not yet been used for two phase flow calculations with surface tension.

Sussman and Puckett [20] proposed a coupled level set and volume of fluid method. Both a level set function as well as the fraction of volume within each cell were used. In this way calculations of curvature could be improved compared to earlier volume of fluid methods. Mass could be conserved almost exactly. Since the fraction of volume is advected, it is however hard to see how any high order of accuracy of the advection can be obtained.

Good introductions to the level set method and more complete reviews can be found in the books by Osher and Fedkiw [10] and by Sethian [15].

1.5 Front Tracking Methods

In front tracking methods the interface is defined explicitly, usually by marker particles distributed on the interface. This approach was used by Unverdi and Tryggvason [23]. The interface is then evolved by solving the ordinary differential equation

$$\frac{dX}{dt} = \vec{u}(X), \quad X \in \Gamma.$$  \hspace{1cm} (1.6)

for each particle. In order to calculate normals and curvatures one needs to keep track of which particle is neighbor to which particle. Particles might move too close together or too far apart, resulting in numerical difficulties. To avoid this, the particles might have to be redistributed along the interface. Since the particles move independently of each other, oscillations in the interface might occur. Some regularizing technique has to be applied to suppress such oscillations. Interpolations also have to be done to interpolate the interface to the fixed grid used for the velocity and pressure. Special care also has to be taken when merging or breakup of interfaces occur.
Another example of a front tracking method is the segment projection method, developed by Tornberg and Engquist [22]. They represented the interface by several line segments. No redistribution of marker points is needed and the calculation of normal and curvature is simpler compared to the method of Unverdi and Tryggvason. The method is however still difficult to implement.

Front tracking methods have turned out to be accurate and thereby to conserve volumes quite well. The implementation does however become complex, in particular for problems in three dimensions.

1.6 Numerical Treatment of Surface Tension and Discontinuities Across the Interface

In general, the density and viscosity are discontinuous across the interface. The surface tension acts only at the interface, creating a jump in the pressure at the interface. When using numerical methods, special care has to be taken to handle these discontinuities. The most simple approach, and which has been used traditionally, is to regularize all jumps and the surface tension. The density and viscosity can be regularized such that instead of jumping discontinuously, they go smoothly from \( \rho_1 \) to \( \rho_2 \) over several grid points.

A regularization of the surface tension was presented by Brackbill, Kothe and Zemach [2]. They suggested to replace the surface tensional force

\[
\vec{F}_{st} = \kappa \hat{n} \delta \Gamma.
\]

by

\[
\tilde{F}_{st} = \kappa \nabla \phi
\]

where \( \phi \) is some function going smoothly from zero to one across the interface. It can easily be verified that as the transition from zero to one of \( \phi \) becomes sharper, \( \tilde{F}_{st} \rightarrow \vec{F}_{st} \). This approach of modeling surface tension have been used for the level set method, volume of fluid methods and front tracking methods. It should be noted that one has to be careful when discretizing regularized delta functions. It was shown by Engquist et al [3] that certain discrete regularized delta functions will produce errors of order 1. This means that errors will persist under grid refinement if an inappropriate approximation of the delta function is used.

More recently, techniques to impose sharp discontinuities at the interface have been developed. For instance, Kang et al. [8] have developed a sharp interface method for two phase flow based on the level set method. Francois et al. [5] present a corresponding method for the volume of fluid method. The sharp approach avoids the unphysical smearing. Because of this, they can be expected to be more accurate than the diffuse methods.
Chapter 2

A Conservative Level Set Method

2.1 Introduction

As we have seen, the level set method is a powerful tool to represent moving interfaces. The main advantage is the simplicity and the possibility to obtain high order of accuracy by using standard techniques for partial differential equations. However, the conservation of mass and volume might be poor. This will in particular be true if the interface is not well resolved or in long time calculations.

The goal of our work has been to construct a method having all the good properties of the level set method. We want to represent the interface as the level set of a globally defined function $\Phi$. The motion of the interface can then be expressed as a partial differential equation. This removes several of the difficulties present in front tracking methods. Furthermore, we want $\Phi$ to be smooth over the interface, so that standard, possibly higher order accurate, methods can be used to solve the PDE and to calculate normal and curvature. This is not the case for volume of fluid methods. All these are properties of standard level set method. In addition to these, we want the method to have some kind of built in conservation. For long time calculations or when the features of the interface are too small to be resolved by the grid, there should still be good conservation of volume and mass.

The method we have developed is described in the two papers included in this thesis. We will briefly summarize the content of these articles in the next sections.

2.2 Summary of Included Papers

Paper 1: A Conservative Level Set Method for Incompressible Two Phase Flow

In Paper 1 a new method to represent and move an interface in a given velocity field $\vec{v}$ is presented. The velocity field is supposed to be divergence free, i.e. $\nabla \cdot \vec{v} = 0$. The interface is represented as the 0.5 contour of a certain regularized characteristic function. This means that $\Phi \approx 1$ on one side of the interface and $\Phi \approx 0$ on the other. Across the interface $\Phi$ goes rapidly from zero to one.

Since the velocity is divergence free, the 0.5 level set can be advected by solving the conservation law

$$\Phi_t + \nabla \cdot (\Phi \vec{v}) = 0. \quad (2.1)$$
Given $\Phi(x, t_n)$ as initial condition, denote the solution of (2.1) at $t = t_{n+1}$ by $\Phi^*(x, t_{n+1})$. As for standard level set methods, the 0.5 contour of $\Phi^*(x, t_{n+1})$ will move correctly, but the shape of $\Phi^*(x, t_{n+1})$ across the interface will be different from that of $\Phi(x, t_n)$. To recover the original profile the PDE

$$\Phi_t + \nabla \cdot (\Phi(1 - \Phi)\hat{n}) = \varepsilon \Delta \Phi$$

(2.2)

$$\Phi(x, 0) = \Phi_0 = \Phi^*(x, t_{n+1})$$

is suggested to be solved to steady state. This gives $\Phi(x, t_{n+1})$. Here, $\varepsilon > 0$ is a small constant and

$$\hat{n} = \frac{\nabla \Phi_0}{|\nabla \Phi_0|}.$$  

The method is different from standard level set method in two important ways. First of all, the reinitialization can be formulated as a conservation law. Because of this, we can use conservative numerical methods to obtain an approximate solution $\Phi_h \approx \Phi$. If there is no flux through the boundaries, this implies that

$$\frac{d}{dt} \left( \int_{\Omega} \Phi_h d\Omega \right) = 0.$$

Secondly, since $\Phi_h$ is an approximate regularized characteristic function, we have that

$$\int_{\Omega} \Phi_h d\Omega \approx A_{\Phi_h=0.5}.$$

Here, $A_{\Phi_h=0.5}$ denotes the area bounded by the interface $\Phi_h = 0.5$. This motivates why we can expect the variations of $A_{\Phi_h=0.5}$ in time to be small.

The method is discretized using conservative second order accurate finite differences. Numerical tests are performed. The convergence of the method for two different velocity fields is checked. These results yield an estimated order of accuracy of two and good conservation of the area bounded by the 0.5 contour. The results are compared with a standard second order level set method. The conservation of the new method is significantly better compared to the standard method. The method is also coupled with a solver of the incompressible two phase flow Navier-Stokes equations with surface tension. Calculations show good conservation, but the convergence rate is low. All calculations are performed on uniform grids in two space dimension.

Paper 2: A Conservative Level Set Method for Incompressible Two Phase Flow II

In Paper 2 the method described in Paper 1 is studied and developed further. First of all, the equation for the reinitialization is modified. In the original formulation (2.2) there was compression only in the direction of $\hat{n}$, while there was diffusion in all directions. This implies that the diffusion in the normal direction will quickly become balanced by the compression. No compression will however balance the diffusion in the tangential direction. This tangential diffusion might thus move the position of the interface. To avoid this, (2.2) is replaced by

$$\Phi_t + \nabla \cdot (\Phi(1 - \Phi)\hat{n}) = \varepsilon \nabla \cdot ((\nabla \Phi \cdot \hat{n}) \hat{n}).$$  

(2.3)

(2.3) gives a diffusive flux only in the direction of $\hat{n}$. 

The convergence of (2.3) to steady state is studied by looking at the corresponding one dimensional problem. The study indicates an exponential convergence on a fast time scale. This suggests that only a few time steps will be needed to reach a steady state of (2.3).

The conservation of $A_{\phi_h=0.5}$ is studied in more detail. We show that, if numerical errors are small, then

$$A_{\phi_h=0.5}(t_n) = A_{\phi_h=0.5}(0) + \delta(t_n),$$

where

$$|\delta(t_n)| < 2 \cdot L_{\Gamma}(t_n)|\kappa(t_n)|_\infty \varepsilon^2.$$ 

Here, $L_{\Gamma}(t_n)$ is the length and $|\kappa(t_n)|_\infty$ the maximum magnitude of the curvature of the interface $\Gamma$ at $t = t_n$.

A finite element discretization is proposed. A simple adaptive procedure based only on the distance to the interface is presented. Calculations of incompressible two phase flow with strong surface tension are presented. The results of the computations show good area conservation as well as second order of accuracy. Calculations were performed for both two and three dimensional problems.
Chapter 3

Open Questions and Future Work

Even though we have been able to use the new conservative level set method with satisfying results, there are still many open questions. Among these are how to handle fluid interfaces in contact with non-slip walls. Another question is how the interface moves during the reinitialization. So far, the method has only been applied to simple problems, where the exact solution is more or less known. A goal is of course to be able to use the method to solve more complicated flow problems. One example is how oil is extracted out of a porous medium by air or water injections. The process of producing hard metals by liquid phase sintering is another interesting problem.

These are all problems we plan to solve in the future. They will be discussed briefly in the following sections.

3.1 Contact Lines

In reality, the interface between two liquids is often in contact with a wall. The intersection of the interface and the boundary is called contact lines. If non-slip boundary conditions are used, i.e., if \( \vec{v} = 0 \) on the boundary, the contact lines will not move. In physical experiments, this is not observed. Instead, interfaces do move along walls. This indicates that the physical model we are using does not model contact lines correctly. It is instead believed that there is a preferred angle between the interface and the wall. Which preferred angle this is, depends on properties of the fluids and the wall. Furthermore, how the interface moves will depend on this angle. We believe that it should be possible to modify our method by imposing some given contact angle. Exactly how this should be done remains to be investigated.

3.2 Analysis of the Method

One important issue that has not yet been analyzed is how the 0.5 contour of \( \Phi \) moves during the reinitialization. It is clear that in the limit \( \varepsilon \to 0 \), the reinitialization will not move the interface at all. But what happens for small \( \varepsilon > 0 \)? If the displacement of the interface is small, for instance, decreases exponentially with \( \varepsilon \), then it is reasonable to use higher order accurate methods to improve the accuracy of the method. However, if the error is \( O(\varepsilon) \), then our method is bound to be low order accurate. An analysis of the motion of the interface during the reinitialization will indicate whether or not our method can be used to move interfaces with high order of accuracy. Furthermore, it will suggest how small \( \varepsilon \) should be chosen to obtain accurate results.

15
3.3 Applications in Petroleum Engineering and Material Science

There are two different physical problems we want to study using our method. To handle both of these, the method must be able to move contact lines correctly.

First of all, we want to simulate the recovery of petroleum by water or gas injections. In general, empirical macroscopic models are used to simulate the flow on a large scale. It is however hard for such models to accurately account for the effect of surface tension and the motion of contact lines. This can be studied using our method. However, only a small volume can be modeled since the geometry of the porous medium is varying on a very small length scale. Results from calculations on the small scale could however be used to improve the macroscopic model. In practice, one is unable to extract all of the oil by water or gas injections. Some oil is always trapped. Simulations could improve the understanding of when such trapping occurs and how it could be avoided.

The second application is the formation of hard metals by liquid phase sintering. Small grains of Tungsten Carbide is mixed with grains of Cobalt. When the powder is heated, the Cobalt melts while the Tungsten Carbide remains solid. The liquid Cobalt tends to pull the solid grains together, creating a densified material. The driving force of this process is believed to be the strong surface tension of liquid Cobalt. When the material is cooled down, a hard solid material is formed. Using numerical simulation it will be possible to increase the understanding of the process. The process of liquid phase sintering is similar to the extraction of oil. An important difference is however that the Tungsten Carbide grains will move and deform slightly. To fully model this, three phases have to be modeled. The conservative level set method for two phase flow thus have to be further developed before it can be used to simulate the complete process of liquid phase sintering.
Bibliography


Paper I
A conservative level set method for two phase flow

Elin Olsson *, Gunilla Kreiss

Department of Numerical Analysis and Computer Science, Royal Institute of Technology, Lindstedsvagen 3, 100 44 Stockholm, Sweden

Received 26 August 2004; received in revised form 8 April 2005; accepted 13 April 2005
Available online 21 June 2005

Abstract

A conservative method of level set type for moving interfaces in divergence free velocity fields is presented. The interface is represented implicitly by the 0.5 level set of a function \( \phi \) being a smeared out Heaviside function, i.e., a function being zero on one side of the interface and one on the other. In a transition layer of finite, constant thickness \( \phi \) goes smoothly from zero to one. The interface is moved implicitly by the advection of \( \phi \), which is split into two steps. First \( \phi \) is advected using a standard numerical method. Then an intermediate step is performed to make sure that the smooth profile of \( \phi \) and the thickness of the transition layer is preserved. Both these steps are performed using conservative second order approximations and thus conserving \( \int \phi \). In this way good conservation of the area bounded by the 0.5 contour of \( \phi \) is obtained.

Numerical tests shows up to second order accuracy and very good conservation of the area bounded by the interface.

The method was also coupled to a Navier–Stokes solver for incompressible two phase flow with surface tension. Results with and without topological changes are presented.

© 2005 Elsevier Inc. All rights reserved.

Keywords: Level set method; Free boundary; Two phase flow; Incompressible flow; Conservative method

1. Introduction

Problems involving moving boundaries and interfaces exist in a wide range of applications, such as multi-phase flow, crystal growth, image processing, front propagations, fluid–structure interactions, etc. Different ways to simulate these problems have been developed. Some of the more commonly used are front tracking methods and level set methods and for incompressible flows volume of fluid methods.
In the simulation of incompressible two phase flow volume of fluid (VOF) methods have been used extensively. In these methods, the interface is given implicitly by a color function, defined to be the fraction of volume within each cell of one of the fluids. From the color function, a reconstruction of the interface is made and the interface is then propagated implicitly by updating the color function. VOF methods are conservative and can deal with topological changes of the interface. However, they are often rather inaccurate, high order of accuracy is hard to achieve because of the discontinuity of the color function. As far as we know, no advection scheme for the volume of fluid method has order higher than two. Also, properties of the interface such as normal and curvature are hard to calculate accurately. Still, the good conservation properties are attractive and quite sophisticated methods have been developed. For one of the early work on VOF, we refer to Noh and Woodward [1] and for a review of this type of methods to Scardovelli and Zaleski [2].

Another approach for free boundary problems is to track the boundary explicitly by markers distributed evenly on the interface, and then propagate the markers. In this way the interface can be represented sharply. This type of methods are often referred to as front tracking methods. Markers may however move close together or far apart, making redistribution of markers necessary. Special care has to be taken to topological changes. Also, if the markers move independently of each other, oscillations in the interface may occur. Another difficulty is the interaction of the interface with a fixed Eulerian grid, which is often needed. All these features makes front tracking methods hard to implement for a general case. A method to simulate multi-fluid flows using front tracking is described in [3].

Lately, level set methods have become popular and have been used in a large variety of applications such as compressible and incompressible two phase flow, image processing and flame propagation, just to mention a few. General descriptions of level set methods can be found in [4,5] and applications to two phase flow in [6,7]. In general, the interface is represented by the zero contour of a signed distance function, the level set function. The movement of the interface is governed by a differential equation for the level set function. The advection is in general done by (weighted) essentially non-oscillatory (WENO, ENO) methods. To keep the level set function a signed distance function, a reinitialization process is needed. Also this process is governed by a differential equation. Level set methods automatically deal with topological changes and it is in general easy to obtain high order of accuracy, just by picking an ENO or WENO scheme with the desired order of accuracy.

One of the drawbacks of level set methods is that they are not conservative. For incompressible two phase flow, loss or gain of mass might occur, which is physically incorrect. The poor mass conservation of level set methods in a finite element framework compared to front tracking methods was pointed out in [8]. Several attempts to improve mass conservation of level set methods have been done. In [9], a combination of the level set method and the VOF method was used in order to obtain the good mass conservation of the VOF method, but using a level set function to obtain better approximations of the curvature. A color function is needed and has to be advected, as in standard volume of fluid schemes. Since this function is discontinuous across the interface special care has to be taken when advecting this function. Due to this, it might be hard to obtain advection schemes of order higher than two without introducing oscillations. The simplicity of the original level set methods is also lost. The problem of mass conservation of level set methods was also addressed in [10]. The authors there propose a hybrid level set – marker particles method to improve accuracy, in particular in underresolved regions. However, in both these cases the original simplicity of level set methods is partly lost.

Our goal is to find an alternative level set function, together with an advection scheme, resulting in conservation of the area (volume in 3 dimensions) bounded by the interface. The velocity field is assumed to be divergence free. To achieve our goal, we use a smeared out Heaviside function as the level set function, i.e., a function being zero in one fluid and one in the other. Over the interface it varies smoothly from zero to one. The advection of the level set function is performed using a conservative scheme with an intermediate
step that keeps the shape and width of the profile across the interface constant. Furthermore, our level set function will be smooth, which makes our method easy to extend to higher order, as opposed to the discontinuous color function.

### 2. Choice of level set function

In standard level set methods, the level set function \( \Phi \) is defined to be a signed distance function

\[
|\Phi(\vec{x})| = d(\vec{x}) = \min_{x_i \in I}(|\vec{x} - \vec{x}_i|),
\]

where \( I \) is the interface, \( \Phi(\vec{x}) > 0 \) on one side of the interface and \( \Phi(\vec{x}) < 0 \) on the other. The advection of \( \Phi \), including a reinitialization step to retain \( \Phi \) as a distance function, is not done in a conservative way, not even for divergence free velocity fields. This implies that the area bounded by the zero level set is not conserved. This is one of the drawbacks of level set methods.

To represent density and viscosity discontinuities over the interface the Heaviside function:

\[
H(\Phi) = \begin{cases} 
0, & \Phi < 0, \\
1, & \Phi > 0
\end{cases}
\]

is needed. In computations, to achieve numerical robustness, a smeared out Heaviside is often used. For example

\[
H_{\text{sm}}(\Phi) = \begin{cases} 
0, & \Phi < -\epsilon, \\
\frac{1}{2} + \frac{\Phi + \sin(\frac{\pi \Phi}{\epsilon})}{2}, & -\epsilon \leq \Phi \leq \epsilon, \\
1, & \Phi > \epsilon,
\end{cases}
\]

where \( \epsilon \) corresponds to half the thickness of the interface.

If we could instead choose

\[
\tilde{\Phi}(\vec{x}) = H_{\text{sm}}(\Phi(\vec{x})),
\]

we would not have to calculate \( H_{\text{sm}} \) from \( \Phi \). More important, assume we have a conservative numerical method to advect \( \tilde{\Phi}(\vec{x}) \) that preserves the smooth profile of \( \Phi(\vec{x}) \). Since the method is conservative \( \int \Phi \) will be conserved exactly. This implies that we can also expect good conservation of \( A_{\Phi=0.5} \), the area bounded by \( \Phi = 0.5 \), since \( A_{\Phi=0.5} \approx \int \Phi \). If we would use the sharp Heaviside, we would preserve the area exactly. However, on a discrete grid, the position of the interface is better approximated by the level set of a smooth function. The interface thickness should therefore typically depend on the grid size, so that the smooth profile can be resolved by the grid.

We can also choose to either define a sharp interface at \( \tilde{\Phi}(\vec{x}) = 0.5 \) or a diffuse interface for \( 0 < \Phi(\vec{x}) < 1 \). In the following, we will refer to our choice of level set function as the phase field function, denoted \( \Phi \).

Normals and curvatures can easily be obtained from our phase field function as:

\[
\hat{n} = \frac{\nabla \Phi}{|\nabla \Phi|},
\]

\[
\kappa = -\nabla \cdot \left( \frac{\nabla \Phi}{|\nabla \Phi|} \right).
\]
3. Advection of the phase field function

3.1. Advection step

We now turn to the problem of finding a method to advect \( \Phi \) that is conservative and that does not change the profile of \( \Phi \) at the interface.

Assume the interface is advected with a given velocity field \( \vec{u} \). This corresponds to the following simple ordinary differential equation for every point \( \vec{x} \) on the interface

\[
\frac{d \vec{x}}{dt} = \vec{u}(\vec{x}).
\]

(2)

As in standard level set methods we can instead solve

\[
\Phi_t + \vec{u} \cdot \nabla \Phi = 0
\]

(3)
on the entire domain. This will move the 0.5 level set of \( \Phi \) according to (2). For incompressible flow, \( \vec{u} \) is always divergence free, i.e., \( \nabla \cdot \vec{u} = 0 \). Eq. (3) is then equivalent to the conservation law

\[
\Phi_t + \nabla \cdot (\Phi \vec{u}) = 0.
\]

When choosing a suitable numerical method to solve this one has to consider:

- The method should be conservative.
- No spurious oscillations should be introduced.
- The thickness of the interface and the profile of \( \Phi \) should be kept constant.

Using a uniform grid, we define grid points \( \vec{x}_{i,j} = (x_i, y_j) = (x_0 + i \Delta x, y_0 + j \Delta y) \) and a grid function \( \Phi_{i,j} \approx \Phi(\vec{x}_{i,j}) \). The velocity, \( \vec{u} = (u, v) \) is assumed to be given on a staggered grid, i.e., \( u \) on grid points \( \vec{x}_{i+\frac{1}{2},j} \) and \( v \) on \( \vec{x}_{i,j+\frac{1}{2}} \). Conservative methods can be written on the form

\[
\frac{d \Phi_{i,j}}{dt} = -\frac{1}{\Delta x} (F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}) - \frac{1}{\Delta y} (G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}}),
\]

(4)

where \( F_{i+\frac{1}{2},j} \) and \( G_{i,j+\frac{1}{2}} \) approximates the flux \( (F, G) = \Phi \vec{u} \) on the staggered grid. Calculating fluxes using centered averaging,

\[
F_{i+\frac{1}{2},j} = 0.5(\Phi_{i,j} + \Phi_{i+1,j})u_{i+\frac{1}{2},j}, \quad G_{i,j+\frac{1}{2}} = 0.5(\Phi_{i,j} + \Phi_{i,j+1})v_{i,j+\frac{1}{2}}
\]

(5)
will typically introduce oscillations close to the interface. This can be seen in Fig. 1(b), which shows results after only a few time steps.

A large amount of work on TVD (total variation diminishing) methods has been done, see [11]. These methods approximate conservation laws without introducing oscillations near discontinuities. Linear TVD methods will however smear discontinuities. A suitable method for the advection of our phase field function would thus be a non-linear TVD method that does not smear discontinuities.

A typical TVD method uses an upwind scheme together with a piecewise linear reconstruction of \( \Phi \). A limiter has to be chosen to determine how the linear reconstruction is made. In time a conservative explicit second order Runge–Kutta discretization can be used. A more detailed description of this method is given in Appendix A. Results using different limiters, i.e., different \( \text{Lim}(x,y) \) in (A.1), are shown in Fig. 1(c)–(f). Here, a rotating velocity field \( (u,v) = (y,-x) \) was used. The figures show the result after one full revolution so that the initial state should be recovered exactly. We used the minmod, van Albada, van Leer and the Superbee limiter. Among TVD schemes, the ones using the Superbee limiter are well known to be the least...
Fig. 1. 0.05, 0.5 and 0.95 Contours of $\Phi$ initially and after one revolution using different numerical methods. (c)-(f) are second order TVD methods, (b) is second order but not TVD. (a) Initial state; (b) centered differences; (c) upwind with Minmod; (d) upwind with Van Albada; (e) upwind with Van Leer; (f) upwind with Superbee.
diffusive, i.e., which will produce the least smearing. It is also clear from Fig. 1(c)–(f) that this limiter preserves the thickness of the interface better than the other limiters.

3.2. Intermediate step

Not even with the Superbee limiter (Fig. 1(f)), the profile and thickness of the interface remains constant. Also, the shape of the profile seems to depend on the normal of the interface relative to the direction of motion. To solve this problem, we add an intermediate step after each time step that will make sure that the interface keeps its thickness and shape. As originally proposed by Harten [12] it is possible to add artificial compression in order to maintain the resolution of contact discontinuities. This can be viewed as an intermediate step where one is solving the conservation law

$$\Phi_t + \nabla \cdot \vec{f}(\Phi) = 0,$$  \hfill (6)

where $\vec{f}$ corresponds to the compressive flux. In our case, we want the artificial compression flux to act in regions where $0 < \Phi < 1$ and in the normal direction of the interface. To achieve this, we choose $\vec{f} = \Phi(1-\Phi)\hat{n}$, where $\hat{n}$ is the normal of the interface. We denote the time variable by $\tau$ to stress that this is an artificial time, not equivalent to the actual time $t$. We note that (6) is a hyperbolic differential equation. As $\tau$ increases, stationary shocks will develop at the interfaces.

To avoid discontinuities at the interface, we add a small amount of viscosity, i.e., we modify the conservation law by

$$\Phi_t + \nabla \cdot \vec{f}(\Phi) = \varepsilon \Delta \Phi,$$  \hfill (7)

or in conservative form

$$\Phi_t + \nabla \cdot \vec{f}(\Phi) = 0$$  \hfill (8)

with

$$\vec{f}(\Phi) = \vec{f}(\Phi) - \varepsilon \nabla \Phi.$$

By solving (8) to steady-state the interface thickness will remain constant and proportional to $\varepsilon$.

This can be approximated in space by

$$\frac{d\Phi_{i,j}}{d\tau} = -\frac{1}{\Delta x} (F_{i+1/2,j} - F_{i-1/2,j}) - \frac{1}{\Delta y} (G_{i,j+1/2} - G_{i,j-1/2}),$$  \hfill (9)

where $F$ and $G$ is the numerical flux at cell faces. We choose:

$$F_{i+1/2,j} = \frac{f(\Phi_{i,j}) + f(\Phi_{i+1,j})}{2} - \varepsilon \frac{\Phi_{i+1,j} - \Phi_{i,j}}{\Delta x},$$

$$G_{i,j+1/2} = \frac{g(\Phi_{i,j}) + g(\Phi_{i,j+1})}{2} - \varepsilon \frac{\Phi_{i,j+1} - \Phi_{i,j}}{\Delta y},$$

with $f$ and $g$ being the $x$ and $y$ components of $\vec{f}$. The interface normal $\hat{n} = \nabla \Phi / |\nabla \Phi|$ is approximated using centered differences:

$$(\nabla \Phi)_{ij} = \frac{\Phi_{i+1,j} - \Phi_{i-1,j}}{2\Delta x} \hat{x} + \frac{\Phi_{i,j+1} - \Phi_{i,j-1}}{2\Delta y} \hat{y},$$

$$\hat{n}_{ij} = \frac{|(\nabla \Phi)_{ij}|}{\nabla \Phi_{ij}},$$
**n_{i,j}** is only calculated once in each intermediate step. That is, after each advection step, \( n_{i,j} \) is calculated and is then kept fixed until steady-state of (8) has been reached. In time we use the same Runge–Kutta approximation as for the advection of \( \Phi \). The method obtained is second order accurate in space and time.

Since we use explicit time stepping we get stability restrictions on \( \Delta t \) due to the viscous term, typically

\[
\Delta t \leq C \frac{(\Delta x)^2}{\varepsilon}.
\]

Experimentally, we found that stability was obtained by choosing \( C = 1/4 \). We choose \( \varepsilon \), which determines the interface thickness, to depend on the grid size in the following way:

\[
\varepsilon = \frac{\left(\Delta x\right)^{1-d}}{2}.
\]

If \( d = 0 \), \( \varepsilon \) and thus the width of the interface, will be proportional to \( \Delta x \). In this case, the resolution of the smooth interface profile will not increase under grid refinement. As will be seen later we do, however, obtain second order accuracy even for \( d = 0 \) in a test calculation of a rotating circle. For a more complicated flow field, as the vortex test in Section 5.2, we had to choose \( d > 0 \) to obtain convergence. However, a small value, \( d = 0.1 \), was enough. In all our calculations we have used

\[
\Delta t = \frac{(\Delta x)^{1+d}}{2}.
\]

As a criteria for steady-state we used

\[
\int |\Phi^{m+1} - \Phi^m| < \text{TOL} \cdot \Delta t
\]

for some specified tolerance TOL. Numerical tests have shown that in practice only a few time steps have to be performed in order to reach steady-state.

Results after one revolution using different methods for the advection together with the artificial compression step are given in Fig. 2. It is clear that we get reasonable solutions for all the advection methods, even the one using centered differences. The thickness of the interface is constant for each case.

### 3.3. Boundary and initial conditions on \( \Phi \)

Appropriate boundary and initial conditions must also be assigned to \( \Phi \). In our calculations, we have either used \( \Phi = 0 \) on the boundaries or assumed the boundaries to be walls with a contact angle of \( \theta = \pi \). In the case of walls, on the boundaries

\[
\hat{n} \cdot \hat{n}_{\text{wall}} = 0
\]

should hold. Here \( \hat{n} \) is the normal of the interface and \( \hat{n}_{\text{wall}} \) is the normal of the wall. This can be transformed to a homogeneous Neumann condition for \( \Phi \)

\[
\hat{n}_{\text{wall}} \cdot \nabla \Phi = 0.
\]

At \( t = 0 \), \( \Phi \) has to be initiated. One way to do this is to set \( \Phi \) to one on one side of the interface and zero on the other. Then (7) is solved numerically until a steady-state has been reached. The resulting function is used as initial data for \( \Phi \). However, in certain cases the steady-state of (7) can be found analytically. For steady-state, \( \nabla \cdot \vec{f}(\Phi) = \varepsilon \Delta \Phi \) should hold and the 0.5 contour of \( \Phi \) should lie along the desired interface. If

\[
\Phi = \left(1 + e^{(\|x - x_c\| - r)/\varepsilon}\right)^{-1},
\]
Fig. 2. Contours of $\Phi$ initially and after one revolution, $t = 2\pi$, using different methods together with the artificial compression step. (a) Initial state; (b) centered differences; (c) Van Albada; (d) upwind with Minmod; (e) upwind with Van Leer; (f) upwind with Superbee.
\[ \nabla \cdot \mathbf{f}(\Phi) = \varepsilon \Delta \Phi \text{ is fulfilled for } x \neq x_c \text{ and the 0.5 contour of } \Phi \text{ corresponds to a circle centered in } \bar{x}_c \text{ with radius } r. \text{ Correspondingly for a horizontal interface at } y = y_{\text{int}} \text{ we can set} \\
\Phi = \left(1 + e^{(y - y_{\text{int}})/r}\right)^{-1} \]

to both fulfill steady-state and obtain the 0.5 contour at \( y = y_{\text{int}} \).

4. Incompressible two phase flow

The next step is to couple our advection method for the phase field function with a two phase incompressible flow solver. We use a diffuse interface model, where surface tension is transformed to a volume force spread over a few layers of cells. The discontinuity in density and viscosity across the interface is also smoothed out.

Surface tension per interfacial area at a point \( \bar{x}_i \) on the interface is given by

\[ F_{\text{sa}}(\bar{x}_i) = \sigma \kappa(\bar{x}_i) \hat{n}(\bar{x}_i). \]

As in [13] choosing a volume force at any point \( \hat{x} \) as

\[ F_{\text{sv}}(\hat{x}) = \sigma \left(-\nabla \cdot \frac{\nabla \Phi}{|\nabla \Phi|}\right) \nabla \Phi \]

will result in the same total force as \( F_{\text{sa}}(\bar{x}_i) \), but spread over the finite interface width. Here, it becomes important that the thickness of the transition layer is kept constant. \( F_{\text{sv}} \) is equal to \( F_{\text{sa}}(\bar{x}_i) \) only in the limit when the thickness of the interface goes to zero. If the interface becomes wide \( F_{\text{sv}} \) will not be a good approximation of \( F_{\text{sa}}(\bar{x}_i) \). Therefore, the interface thickness should not become too wide. A too sharp transition, on the other hand, will yield difficulties in numerically computing \( F_{\text{sv}} \) accurately, since up to second order derivatives of \( \Phi \) are needed.

The non-dimensionalized incompressible Navier–Stokes equations with surface tension and gravity are:

\[
\begin{align*}
\nabla \cdot \mathbf{u} &= 0, \\
\tilde{u}_i + (\tilde{u} \cdot \nabla) \tilde{u} &= -\nabla p + \frac{1}{\rho} \nabla \cdot \left( \mu \left( \nabla \tilde{u} + (\nabla \tilde{u})^T \right) \right) + \frac{1}{F_{\text{Fr}}^2} \tilde{e}_g + \frac{1}{\rho We} \tilde{F}_{\text{sv}}, \\
\Phi_i + \tilde{u} \cdot \nabla \Phi &= 0,
\end{align*}
\]

where \( Re = \frac{\rho_{\text{ref}} u_{\text{ref}} l_{\text{ref}}}{\nu_{\text{ref}}} \) is the Reynolds number, \( Fr = \frac{u_{\text{ref}}}{\sqrt{\frac{\gamma_{\text{ref}}}{\rho_{\text{ref}}} g l_{\text{ref}}}} \) the Froude number, \( We = \frac{\rho_{\text{ref}} u_{\text{ref}}^2 l_{\text{ref}}}{\sigma} \) the Weber number and \( \tilde{F}_{\text{sv}} = \tilde{F}_{\text{sv}}/\sigma, \tilde{e}_g \) is the unit vector in the direction of gravitation, and \( \rho \) and \( \mu \) the non-dimensionalized density and viscosity, respectively. The density and viscosity varies smoothly over the interface by letting:

\[
\begin{align*}
\rho &= \rho_1 + (\rho_2 - \rho_1) \cdot \Phi, \\
\mu &= \mu_1 + (\mu_2 - \mu_1) \cdot \Phi
\end{align*}
\]

with \( \rho_1, \rho_2 \) and \( \mu_1, \mu_2 \) being the dimensionless densities and viscosities of the two fluids, respectively.

We use a staggered grid for the discretization, i.e., \( \Phi \) and \( p \) are given at grid points \( \bar{x}_{i,j} \), whereas \( u \) is given on \( \bar{x}_{i+1/2,j} \) and \( v \) on \( \bar{x}_{i,j+1/2} \). Eqs. (10) and (11) are solved numerically using extensions of the Marker and Cell method [14] (see Appendix B for a thorough description) together with our conservative shape preserving advection scheme for the solution of (12). The pressure is updated implicitly by solving a Poisson equation with variable coefficients. This linear system is solved by a direct banded solver. The velocity is then
updated explicitly and finally \( \Phi \) is advected with the calculated velocity. The method is second order accurate in space and first order in time.

Note that since we use conservative schemes for updating \( \Phi_{i,j} \), \( \int \Phi_{i,j} \Delta x \Delta y \) is conserved exactly. Since we define the density by (13), also \( \int \rho_{i,j} \Delta x \Delta y \) will be conserved. Mass is thus conserved exactly by our method.

5. Results

We tested our method on four different cases. First, two tests using our advection scheme for given divergence free velocity fields were investigated. Convergence studies were performed, including estimates of order of accuracy. Then, the advection was coupled to the Navier–Stokes solver described in Section 4 for the case of a rising air bubble in water. Finally, a test case involving topological changes of the interface was considered through a water droplet falling through air and then hitting a water surface.

In the two last simulations, the velocity component normal to the walls were set to zero on the boundary. For the tangential component homogeneous Neumann conditions were used. Homogeneous Neumann conditions were also used for \( U \) on all boundaries, i.e., a contact angle of \( \pi \) degrees. From these boundary conditions and the Navier–Stokes equations, Neumann conditions on the pressure follows, \( \partial p / \partial x = 0 \) on the vertical boundaries and \( \partial p / \partial y = \rho / F r^2 \) on the horizontal ones.

5.1. Rotating circle

To test our advection scheme a circle was rotated in the constant velocity field \( (u,v) = (v,-x) \). Solutions on different grids after one revolution, i.e., at \( t = 2\pi \), were compared. The viscosity parameter in (7) was set to \( \nu = \Delta x / 2 \), so that the thickness of the transition layer gets smaller and smaller as the mesh becomes finer. Contour lines corresponding to \( \Phi = 0.05, 0.5 \) and \( 0.95 \) of the solutions at \( t = 2\pi \) on four different grids, \( \Delta x = 0.08, 0.04, 0.02 \) and \( 0.01 \), with \( \Delta t = \Delta \tau = \Delta x / 2 \) are shown in Fig. 3(a). Four artificial compression steps were performed after each time step. On the boundary we used \( \Phi = 0 \).

The conservation of the area within the 0.5 contour is shown in Fig. 3(b). Even though there is a small variation of the area, there is no drift, i.e., the maximum deviation from the initial area does not increase

\( \begin{align*}
  &\text{Fig. 3. Results on the different grids, upwind scheme with Superbee: (a) contour lines corresponding to } \Phi = 0.05, 0.5 \text{ and } 0.95 \text{ at } t = 2\pi; \\
  &\text{(b) area conservation.}
\end{align*} \)
with time. On the coarsest grid we get a maximum deviation less than 0.5% and on the finest only 0.035%. The difference in initial area for the different grids is due to the discretization error of the initial \( \Phi \). Clearly from the result this error is of order \((\Delta x)^2\).

From the result, we estimated the order of accuracy with respect to the position of the circle and with respect to the error measured by

\[
\int |H(\Phi_{\text{numerical}}) - H(\Phi_{\text{exact}})|/L
\]

as was done in [7], where:

\[
H(\Phi) = 0, \quad \Phi < 0.5, \tag{15}
\]

\[
H(\Phi) = 1, \quad \Phi > 0.5 \tag{16}
\]

and \( L \) the perimeter size. In this way, we can measure the error of the sharp interface defined by \( \Phi = 0.5 \). The position of the bubble was defined by the center of mass:

\[
X_{\text{center}} = \frac{\sum_{i,j} \Phi_{i,j} \cdot x_i}{\sum_{i,j} \Phi_{i,j}}, \quad Y_{\text{center}} = \frac{\sum_{i,j} \Phi_{i,j} \cdot y_j}{\sum_{i,j} \Phi_{i,j}}.
\]

Calculated orders of accuracy are shown in the following table:

<table>
<thead>
<tr>
<th>( \Delta x )</th>
<th>Order (( H(\Phi) ))</th>
<th>Order (( y)-pos)</th>
<th>Order (( x)-pos)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.04</td>
<td>1.4</td>
<td>2.1</td>
<td>1.2</td>
</tr>
<tr>
<td>0.02</td>
<td>1.2</td>
<td>2.9</td>
<td>1.5</td>
</tr>
<tr>
<td>0.01</td>
<td>1.2</td>
<td>2.4</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Fig. 4. Results on the different grids, central scheme: (a) contour lines corresponding to \( \Phi = 0.05, \Phi = 0.5 \) and \( \Phi = 0.95 \) at \( t = 2\pi \); (b) area conservation.
In the previous calculations, we used the upwind scheme together with the piecewise linear reconstruction defined by the Superbee limiter. Exactly the same calculations were also done using the central scheme, i.e., (4) with (5). The obtained results are shown in Fig. 4(a) and (b).

The result is very similar to the result obtained with the Superbee scheme. Contours corresponding to the 0.5 level set of $\Phi$ are shown for each method in Fig. 5(a) and (b).

The following orders of accuracy were obtained:

<table>
<thead>
<tr>
<th>$\Delta x$</th>
<th>Order ($H(\Phi)$)</th>
<th>Order (y-pos)</th>
<th>Order (x-pos)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.04</td>
<td>2.0</td>
<td>2.2</td>
<td>2.0</td>
</tr>
<tr>
<td>0.02</td>
<td>1.8</td>
<td>2.1</td>
<td>1.9</td>
</tr>
<tr>
<td>0.01</td>
<td>2.0</td>
<td>2.0</td>
<td>1.9</td>
</tr>
</tbody>
</table>

We thus obtain higher order of accuracy for the central scheme, even though on the coarsest grid, the upwind scheme performs better.

5.2. Vortex test

To test our advection scheme on a more complicated flow, we used the following velocity field on the unit square:

\[
\begin{align*}
    u &= \sin^2(\pi x) \sin(2\pi y), \\
    v &= -\sin^2(\pi y) \sin(2\pi x).
\end{align*}
\]

A circle with radius 0.15 centered at (0.5, 0.75) was used as initial condition. At $t = T$ the flow field was reversed, so that the exact solution at $t = 2T$ should coincide with the initial condition. To obtain convergence, we had to choose $\varepsilon = (\Delta x)^{0.9}/2$. Computations on four grids ($32^2$, $64^2$, $128^2$, $256^2$) were performed for

![Fig. 5. 0.5 Contours of $\Phi$, using different grids and different numerical methods: (a) upwind with Superbee; (b) central scheme.](image-url)
$T = 1$ and $T = 0.5$. The solution for $T = 1$ at $t = 0$, $t = 0.5$, $t = 1$ and $t = 2$ are shown in Fig. 6. The conservation of the area bounded by the 0.5 contour, $T = 1$, is given in Fig. 7(a) and the 0.5 contours of $\Phi$ at $t = 2$ in Fig. 7(b). The corresponding results for $T = 0.5$ are shown in Fig. 8(a) and (b).

The results can be compared with results using standard level set method with reinitialization, see Figs. 9(a) and (b) and 10(a) and (b). Second order ENO schemes were used for both advection and reinitialization, cf. [6].

In Fig. 6, we note a pinch off as the thickness of the stretched circle gets close to the thickness of the interface. This is a numerical effect that can only be avoided if the thickness of the interface is smaller than the distance between two interfaces. We also note from Fig. 7(a) that this pinch off results in a small temporary mass loss. This mass loss is however small compared to the standard level set method, and the mass is recovered at $t = 2T$. For $T = 0.5$, the interface remains well resolved and the mass conservation is very good (Fig. 8(a)). Our method is clearly better with respect to mass conservation compared to the standard method. (Note the difference in scaling in Figs. 7(a), 9(a), 8(a) and 10(a).) This holds independently of whether the interface is well resolved or not.

As for the rotating bubble in previous section, we estimate the order of accuracy at $t = 2T$ with respect to the error defined by (14):

<table>
<thead>
<tr>
<th>$\Delta x$</th>
<th>Error ($T = 1$)</th>
<th>Order ($T = 1$)</th>
<th>Error ($T = 0.5$)</th>
<th>Order ($T = 0.5$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/36</td>
<td>0.049</td>
<td></td>
<td>0.029</td>
<td></td>
</tr>
<tr>
<td>1/62</td>
<td>0.025</td>
<td>0.97</td>
<td>0.0050</td>
<td>2.6</td>
</tr>
<tr>
<td>1/128</td>
<td>0.0023</td>
<td>3.5</td>
<td>0.0012</td>
<td>2.1</td>
</tr>
<tr>
<td>1/256</td>
<td>$7.2 \times 10^{-4}$</td>
<td>1.7</td>
<td>$4.7 \times 10^{-4}$</td>
<td>1.3</td>
</tr>
</tbody>
</table>

Fig. 6. Vortex test on four different grids at $t = 0$, $t = 0.5$, $t = 1$, $t = 2$. $T = 1$. 

Finally, we performed a non-reversed simulation on the finest grid until $t = 4$. The solution at $t = 1$, $t = 2$, $t = 3$ and $t = 4$ is shown in Fig. 11. Clearly, the grid is not fine enough to resolve the interface.

5.3. Rising bubble

An air bubble in water initially at rest was studied. The reference density and viscosity was set to the density and viscosity of water: $\rho_{\text{ref}} = 1.0 \times 10^3 \text{ kg/m}^3$, $\mu_{\text{ref}} = 1.0 \times 10^{-3} \text{ N s/m}^2$ so that $\rho_1 = 1$, $\rho_2 = 0.0013$, $\mu_1 = 1$, and $\mu_2 = 0.016$. Letting $\sigma = 7.3 \times 10^{-2} \text{ N/m}$, $l_{\text{ref}} = 5.0 \times 10^{-3} \text{ m}$ and $u_{\text{ref}} = 0.1 \text{ m/s}$, we obtain $Re = 500$, $Fr = 0.45$ and $We = 0.68$. Again, the interface thickness was determined by $\varepsilon = (\Delta x)^{0.9}/2$. We chose the time step by stability with respect to viscous and convective terms.
\[
\Delta t = \frac{1}{\left( \frac{\text{max}(|u|)}{\Delta x} + \frac{\text{max}(|v|)}{\Delta y} + \frac{4}{\min(\rho \cdot \text{Re}/\mu)} \left( \frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} \right) \right)}.
\]

(17)

Results at \( t = 0.5 \) on the different grids, \( \Delta x = 2/25, \Delta x = 2/50, \Delta x = 2/100 \) and \( \Delta x = 2/200 \) are shown in Fig. 12. The time evolution of the area bounded by the 0.5 contour of \( \Phi \) is shown in Fig. 13(a). The conservation is obviously very good. Even on the coarsest grid the area fluctuation is only about 0.1%. As in the other
tests, the difference in initial area is due to the discretization error of the initial $\Phi$. The velocity of center of mass is shown in Fig. 13(b) and the 0.5 contours of $\Phi$ in Fig. 14. The solid line corresponds to $\Delta x = 2/200$, the dashed to $\Delta x = 2/100$, the dashed-dotted to $\Delta x = 2/50$ and the dotted line to $\Delta x = 2/25$. We observe convergence of both the contours and the velocity, although the rate of convergence is rather slow. A possible reason for this can be the smearing of surface tension, viscosity and density.
Fig. 15. Initial state of falling droplet.

Fig. 16. Falling droplet from $t = 0$ to $t = 3.5$, 0.5 contour of $\Phi$. 
Fig. 17. Falling droplet, close to surface.

Fig. 18. Falling droplet, close to surface. 0.05, 0.5 and 0.95 contours of $\Phi$. 
5.4. Falling droplet

Finally, a problem involving topological changes of the interface was studied: a small droplet of water falls through air until it hits a water surface. The initial state is shown in Fig. 15.

As in the previous section, we set $q_{\text{ref}} = 1.0 \times 10^3 \text{ kg/m}^3$, $l_{\text{ref}} = 1.0 \times 10^3 \text{ N m}^2$, $\rho_1 = 1$, $\rho_2 = 0.0013$, $\mu_1 = 1$, and $\mu_2 = 0.016$ and $\sigma = 7.3 \times 10^{-2} \text{ N/m}$. By choosing $l_{\text{ref}} = 1.0 \times 10^{-3} \text{ m}$ and $u_{\text{ref}} = 1.0 \times 10^{-2} \text{ m/s}$, we get $Re = 10$, $Fr = 0.10$ and $We = 0.0014$. The discretization parameters were chosen as $\Delta x = 0.06$, $\Delta t$ by stability as in (17).

Results from a droplet falling is shown in Figs. 16 and 17. We see that in the air the droplet remains quite circular. This is expected since on this small length scale surface tension is large. The horizontal surface stays straight as the droplet is approaching. One might expect that the air under the droplet would create a small bump on the horizontal surface. However, on this small length scale these effects are too small to be apparent in the results. We also note that as the distance of the bubble and the surface becomes close to the thickness of the interface, the droplet slightly attracts the surface. This is a numerical effect due to the diffuse representation of interfaces. As the droplet hits the water, waves propagating towards the wall are generated. Finally the waves are damped. We see that our method has no problem dealing with the topological change of the interface. In Fig. 18, the three contour lines corresponding to $\Phi = 0.05$, $\Phi = 0.5$ and $\Phi = 0.95$ are shown. We note that the interface keeps it thickness even as the droplet hits the surface.

6. Conclusions

We have constructed a numerical method for the advection of an interface in a divergence free velocity field. The method is conservative and the thickness of the diffuse interface is kept constant. Our method is easy to implement and the extension to three dimensions is straight forward. No special care has to be taken concerning topology changes, since this is automatically incorporated in the method. We have used second order approximations and numerical tests have also shown an actual order of accuracy of about two. Mass conservation is significantly better compared to the standard level set method.

Since our method is based on a certain smooth level set function together with a set of differential equations, other numerical methods can easily be applied. For example, higher order accurate and/or finite element discretization could be used. In this way, it should be possible to construct a conservative method with order greater than two.

Acknowledgments

The authors thank Gustav Amberg, Department of Mechanics, KTH, for helpful discussions on fluid dynamics. This work was supported by SSF Grant A3 02:123.

Appendix A. TVD method for advection

The TVD method for the advection of $\Phi$ using the upwind scheme together with a piecewise linear reconstruction can be summarized as follows.

A piecewise linear reconstruction of $\Phi$ is made. On each cell we have

$$\Phi(x,y) = \Phi_{ij} + s_{ij}^x(x - x_{ij}) + s_{ij}^y(y - y_{ij}).$$
The slopes $s_{i,j}^x$ and $s_{i,j}^y$ are calculated by:

\begin{align}
{s_{i,j}^x} &= \text{Lim} \left( \frac{\Phi_{i+1,j} - \Phi_{i,j}}{\Delta x}, \frac{\Phi_{i,j} - \Phi_{i-1,j}}{\Delta x} \right), \\
{s_{i,j}^y} &= \text{Lim} \left( \frac{\Phi_{i,j+1} - \Phi_{i,j}}{\Delta y}, \frac{\Phi_{i,j} - \Phi_{i,j-1}}{\Delta y} \right),
\end{align}

where $\text{Lim}(x,y)$ defines the limiter. The Superbee limiter is defined as

\[
\text{Lim}(x,y) = \begin{cases} 
\text{sign}(x) \max(|x|, |y|) & \text{if } |x|/2 \leq |y| \leq 2|x| \text{ and } xy > 0, \\
2\text{sign}(x) \min(|x|, |y|) & \text{if } |x|/2 \geq |y| \text{ or } |y| \geq 2|x| \text{ and } xy > 0, \\
0 & \text{if } xy < 0.
\end{cases}
\]

For the definition of other limiter, e.g. the minmod limiter, see for example [11].

The evolution of $\Phi_{i,j}$ is given by

\[
\frac{d\Phi_{i,j}}{dt} = -\frac{1}{\Delta x} (F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}) - \frac{1}{\Delta y} (G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}}),
\]

where the fluxes are approximated using the upwind scheme for the linear reconstruction defined by the Superbee limiter:

\[
F_{i+\frac{1}{2},j} = \max(u_{i+\frac{1}{2},j}, 0) \Phi_{i+\frac{1}{2},j}^- + \min(u_{i+\frac{1}{2},j}, 0) \Phi_{i+\frac{1}{2},j}^+,
\]
\[
G_{i,j+\frac{1}{2}} = \max(v_{i,j+\frac{1}{2}}, 0) \Phi_{i,j+\frac{1}{2}}^- + \min(v_{i,j+\frac{1}{2}}, 0) \Phi_{i,j+\frac{1}{2}}^+.
\]

Here

\[
\Phi_{i+1,j}^- = \Phi_{i,j} + \frac{\Delta x}{2} s_{i,j}^x, \quad \Phi_{i,j+1}^- = \Phi_{i,j} + \frac{\Delta y}{2} s_{i,j}^y, \\
\Phi_{i+1,j}^+ = \Phi_{i+1,j} - \frac{\Delta x}{2} s_{i+1,j}^x, \quad \Phi_{i,j+1}^+ = \Phi_{i,j+1} - \frac{\Delta y}{2} s_{i,j+1}^y.
\]

Finally, we discretize in time using an explicit second order TVD Runge–Kutta. We rewrite the system of ODE:s in (A.3) as

\[
\frac{d\tilde{\Phi}}{dt} = \mathcal{F}(\tilde{\Phi}).
\]

$\tilde{\Phi}$ is a vector containing all of the grid values $\Phi_{i,j}$. The second order Runge–Kutta we used is defined by:

\[
\tilde{\Phi}^* = \tilde{\Phi}^n + \Delta t \mathcal{F}(\tilde{\Phi}^n), \\
\tilde{\Phi}^{**} = \tilde{\Phi}^* + \Delta t \mathcal{F}(\tilde{\Phi}^*), \\
\tilde{\Phi}^{n+1} = \frac{1}{2}(\tilde{\Phi}^n + \tilde{\Phi}^{**}).
\]

Appendix B. Discretization of the Navier–Stokes equations

The incompressible Navier–Stokes equations we are solving are given by:

\[
\nabla \cdot \bar{u} = 0,
\]
\[
\bar{u}_i + (\bar{u} \cdot \nabla)\bar{u} = -\frac{\nabla p}{\rho} + \frac{1}{\rho \text{Re}} \nabla \cdot \left( \mu \left( \nabla \bar{u} + (\nabla \bar{u})^T \right) \right) + \frac{1}{\rho \text{Fr}^2} \bar{e}_g + \frac{1}{\rho \text{We}} \bar{F}_{sv}.
\]
We discretize using an extended version of the method described in [14]:

\[
\begin{align*}
    u_{i,j}^{n+1} &= u_{i,j}^n + Q_{i,j}^{n+1} \frac{\Delta t}{\rho_{i,j}^n} (p_{i,j+1}^{n+1} - p_{i,j}^{n+1}), \\
    v_{i,j}^{n+1} &= v_{i,j}^n + R_{i,j+1}^{n} \frac{\Delta t}{\rho_{i,j}^n} (p_{i,j+1}^{n+1} - p_{i,j}^{n+1}),
\end{align*}
\]

with:

\[
\begin{align*}
    Q_{i,j}^{n} &= \Delta t \left( -u u_x + v u_y + \frac{(2 \mu_x) x + (\mu(u_x + v_x))_x}{\rho \text{Re}} + \frac{1}{\rho \text{We}} F_x^x \right)_{i,j}^{n}, \\
    R_{i,j}^{n} &= \Delta t \left( -u v_x - v v_y + \frac{(2 \mu_y) y + (\mu(u_y + v_y))_y}{\rho \text{Re}} + \frac{1}{\rho \text{We}} F_y^y \right)_{i,j}^{n}.
\end{align*}
\]

The discretization of the surface tension is done in the following way. The divergence of \( \hat{n} \) is first approximated as:

\[
\left( \nabla \cdot \frac{\nabla \Phi}{|\nabla \Phi|} \right)_{i,j} = \frac{1}{2\Delta x} \left( \frac{(\Phi_x)_{i+1,j} - (\Phi_x)_{i-1,j}}{|(\nabla \Phi)_{i+1,j}|} - \frac{(\Phi_x)_{i-1,j} - (\Phi_x)_{i+1,j}}{|(\nabla \Phi)_{i-1,j}|} \right) + \frac{1}{2\Delta y} \left( \frac{(\Phi_y)_{i,j+1} - (\Phi_y)_{i,j-1}}{|(\nabla \Phi)_{i,j+1}|} - \frac{(\Phi_y)_{i,j-1} - (\Phi_y)_{i,j+1}}{|(\nabla \Phi)_{i,j-1}|} \right),
\]

where the gradient of \( \Phi \) is calculated using

\[
(\nabla \Phi)_{i,j} = (\Phi_x)_{i,j} \hat{x} + (\Phi_y)_{i,j} \hat{y} = \frac{\Phi_{i+1,j} - \Phi_{i-1,j}}{2\Delta x} \hat{x} + \frac{\Phi_{i,j+1} - \Phi_{i,j-1}}{2\Delta y} \hat{y},
\]

and

\[
|\nabla \Phi|_{i,j} = \sqrt{(\Phi_x)^2_{i,j} + (\Phi_y)^2_{i,j}}.
\]

Finally, we can calculate the surface tension:

\[
\begin{align*}
    (F_x^x)_{i,j+1} &= \frac{1}{2} \left( \left( \nabla \cdot \frac{\nabla \Phi}{|\nabla \Phi|} \right)_{i,j} (\Phi_x)_{i,j} + \left( \nabla \cdot \frac{\nabla \Phi}{|\nabla \Phi|} \right)_{i+1,j} (\Phi_x)_{i+1,j} \right), \\
    (F_y^y)_{i,j+1} &= \frac{1}{2} \left( \left( \nabla \cdot \frac{\nabla \Phi}{|\nabla \Phi|} \right)_{i,j} (\Phi_y)_{i,j} + \left( \nabla \cdot \frac{\nabla \Phi}{|\nabla \Phi|} \right)_{i+1,j} (\Phi_y)_{i+1,j} \right).
\end{align*}
\]

The other terms of \( Q_{i,j}^{n} \) are discretized by:

\[
\begin{align*}
    (u u_x)_{i,j+1/2} &= u_{i+1/2,j} - u_{i-1/2,j}, \\
    (v u_y)_{i,j+1/2} &= \frac{1}{4} \left( v_{i,j+1} + v_{i,j-1} + v_{i+1,j+1/2} + v_{i+1,j-1/2} \right) \cdot \frac{u_{i+1/2,j+1} - u_{i+1/2,j-1}}{2\Delta y}, \\
    (\mu u_x)_{i,j+1/2} &= \frac{\mu_{i+1,j}(u_{i+1/2,j} - u_{i-1/2,j}) - \mu_{i,j}(u_{i+1/2,j} - u_{i-1/2,j})}{(\Delta x)^2}, \\
    (\mu u_y)_{i+1/2,j} &= \frac{\mu_{i+1/2,j+1}(u_{i+1/2,j+1} - u_{i+1/2,j-1}) - \mu_{i+1/2,j-1}(u_{i+1/2,j} - u_{i+1/2,j-1})}{(\Delta y)^2}, \\
    (\mu v_x)_{i,j+1/2} &= \frac{\mu_{i+1/2,j}(v_{i+1,j+1/2} - v_{i+1,j-1/2}) - \mu_{i+1/2,j-1}(v_{i+1,j+1/2} - v_{i+1,j-1/2})}{\Delta x \Delta y}.
\end{align*}
\]
To calculate \( \rho \) and \( \mu \) at cell faces we use:
\[
\begin{align*}
(\rho)_{i,j} &= \rho_1 + (\rho_2 - \rho_1) \cdot \Phi_{i,j}, \\
(\rho)_{i,j+\frac{1}{2}} &= \frac{1}{2}((\rho)_{i,j} + (\rho)_{i,j+1}), \\
(\rho)_{i+\frac{1}{2},j} &= \frac{1}{2}((\rho)_{i,j} + (\rho)_{i+1,j})
\end{align*}
\]

and, respectively, for \( \mu \).

We require \( \vec{u} \) to be divergence free at \( t^{n+1} \), i.e., \( (\nabla \cdot \vec{u})_{i,j}^{n+1} = 0 \), where
\[
(\nabla \cdot \vec{u})_{i,j}^{n+1} = \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x} + \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta y}.
\]

This yields the pressure implicitly as
\[
\left( \nabla \cdot \left( \frac{\nabla p^{n+1}}{\rho^n} \right) \right)_{i,j} = (Q_x + R_y)_{i,j}^n,
\]
which is discretized as
\[
\frac{1}{(\Delta x)^2} \left( \frac{p_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} - p_{i-\frac{1}{2},j}^{n+1}}{\rho_{i+\frac{1}{2},j}^n} - \frac{p_{i+\frac{1}{2},j}^{n+1} - p_{i-\frac{1}{2},j-\frac{1}{2}}^{n+1}}{\rho_{i-\frac{1}{2},j}^n} \right) + \frac{1}{(\Delta y)^2} \left( \frac{p_{i,j+\frac{1}{2}}^{n+1} - p_{i-\frac{1}{2},j}^{n+1}}{\rho_{i,j+\frac{1}{2}}^n} - \frac{p_{i-\frac{1}{2},j}^{n+1} - p_{i-\frac{1}{2},j-\frac{1}{2}}^{n+1}}{\rho_{i-\frac{1}{2},j}^n} \right) = \left( \frac{Q_{x, i+\frac{1}{2},j}^{n} - Q_{x, i, j-\frac{1}{2}}^{n}}{\Delta x} + \frac{R_{y, i, j+\frac{1}{2}}^{n} - R_{y, i-\frac{1}{2},j}^{n}}{\Delta y} \right).
\]

This linear system was solved using a direct solver for banded matrices. Alternatively some iterative method such as a preconditioned conjugate gradient method or a multigrid method could have been used.

References

Paper II
A conservative level set method for two phase flow II

Elin Olsson *, Gunilla Kreiss

Department of Numerical Analysis and Computer Science
Royal Institute of Technology
100 44 Stockholm, Sweden

Abstract

In this paper we continue to develop and study the conservative level set method for incompressible two phase flow with surface tension introduced in [J. Comput. Phys. 210 (2005) 225-246]. We formulate a modification of the reinitialization and present a theoretical study of what kind of conservation we can expect of the method. A finite element discretization is presented as well as an adaptive mesh control procedure. Numerical experiments relevant for problems in petroleum engineering and material science are presented. For these problems the surface tension is strong and conservation of mass is important. Problems in both two and three dimensions with uniform as well as non uniform grids are studied. From these calculations convergence and conservation is studied. Good conservation and convergence are observed.

Key words: Level set method, Two phase flow, Conservative method

* Corresponding author

Email addresses: elino@kth.se (Elin Olsson), gunillak@nada.kth.se (Gunilla Kreiss).

Preprint submitted to Elsevier Science 10 January 2006
1 Introduction

To study the extraction of petroleum and the construction of hard metals through liquid phase sintering it is necessary to be able to model incompressible two phase flow with surface tension accurately. If no phase change occurs the mass of each of the two fluids should be conserved. Since the flow is incompressible this implies that the volume occupied by any of the fluids should be preserved as well. It is therefore of importance that the method used to simulate the flow conserves these volumes.

The most commonly used methods to model incompressible two phase flow with surface tension are the volume of fluid method (VOF)[1] [2], the level set method [3] [4], the front tracking method [5] and phase field methods [6]. Here we focus on the volume of fluid method and the level set method. There are advantages and disadvantages of both of these methods. The VOF method has the main advantage of conserving the volumes of the two fluids exactly. The interface is however represented by a discontinuity of a globally defined function. Because of the discontinuity it is hard both to move the interface as well as calculating surface tension, which depends on the mean curvature of the interface, accurately. Specially designed methods have to be used to advect the interface. Standard finite element discretizations will for example not work.

In level set methods the interface is defined by the zero contour of a signed distance function, the level set function. Since this function is smooth across the interface it is easier to advect the interface as well as to calculate the curvature with high order of accuracy. Both finite element and finite difference approximations have been used sucessfully to advect the interface. There is however no built in volume conservation. A small amount of mass is thus lost or gained in each time step. As time evolves these errors will typically accumulate.

In [7] we constructed a modified level set method with built in conservation. Here, the level set function \( \Phi \) was a regularized characteristic function. A reinitialization procedure, formulated as a conservation law, was used to preserve the smooth profile of the regularized characteristic function. Conservative methods could then be used for both advection and reinitialization. Because of this \( \int_{\Omega} \Phi d\Omega \) was conserved exactly. Since \( \Phi \) was a regularized characteristic function, this implied a good conservation of the volume bounded by the sharp interface \( \Phi = 0.5 \). The numerical calculations showed very good conservation of the volume bounded by the sharp interface. For the test cases with given velocity fields good convergence was obtained. A rising air bubble in water was also studied. Mass conservation was very good, but the convergence as the grid was refined was slow.
We believe that a possible reason for the slow convergence for the rising bubble was the formulation of the reinitialization. In the reinitialization there was diffusion. The diffusion in the direction normal to the interface was balanced by a compressive term, such that the resulting motion of the interface could be expected to be small. The diffusion tangential to the interface was however not balanced by any compression. This tangential diffusion might thus have moved the interface slightly. In this paper we propose the diffusion in the reinitialization to be only in the direction normal to the interface. As will be seen we obtain good convergence with an estimated order of accuracy of two for two phase flow calculations with strong surface tension.

The convergence to steady state in the reinitialization step will be studied. We will see that we can expect this convergence to be exponential in time on a fast time scale. This implies that we can expect the computational expenses due to the reinitialization to be small. We will also perform a theoretical analysis on what kind of conservation we can expect of the method. The numerical tests agree well with our theoretical results.

We also propose a finite element discretization of the method. An adaptive procedure is also presented, such that the grid can be refined close to the interface. Using an existing finite element package this could easily be implemented for both two and three dimensional problems. Results for a few different problems are presented to investigate the conservation and convergence of the method.

2 The conservative level set method

A conservative level set method for motion of interfaces in a given divergence free velocity field was introduced in [7]. We summarize the basic idea of the method here. Assume an arbitrary domain \( \Omega \) divided into subdomains \( \Omega_1 \) and \( \Omega_2 \) such that \( \Omega_2 = \Omega \setminus \Omega_1 \). Let \( \Gamma \) be the internal boundary between \( \Omega_1 \) and \( \Omega_2 \). The idea of the conservative level set method is to define the boundary \( \Gamma \) implicitly by a function \( \Phi \) being a regularized characteristic function. This means that \( \Phi \) goes rapidly from zero to one across the interface \( \Gamma \) and that \( \Phi \approx 1 \) for \( x \in \Omega_1 \) and \( \Phi \approx 0 \) for \( x \in \Omega_2 \) away from the boundary \( \Gamma \). \( \Gamma \) can be sharply defined as the 0.5 contour of \( \Phi \). The normal, \( \hat{n} \), and curvature, \( \kappa \), of the interface can easily be calculated from \( \hat{n} = \nabla \Phi / |\nabla \Phi| \) and \( \kappa = -\nabla \cdot \hat{n} \). As the interface moves we want to keep the shape of the profile constant. Small perturbations in the shape should be damped, since in numerical computations there will always be perturbations. How the shape is preserved will be motivated by a study of the one dimensional case in the following section.
2.1 Stable traveling wave solutions in one dimension

Consider advection in one dimension for some constant velocity $v$:

$$\Phi_t + v\Phi_x = 0. \quad (1)$$

Assume that initially $\Phi$ is the regularized characteristic function given by

$$\Phi_0(x) = \frac{1}{1 + e^{-x/\varepsilon}}. \quad (2)$$

Next assume an initial perturbation $\delta(x)$, i.e.

$$\Phi(x,0) = \Phi_0(x) + \delta(x)$$

The solution to (1) is then

$$\Phi(x,t) = \Phi_0(x - vt) + \delta(x - vt).$$

Perturbations will thus not be damped but simply advected with $v$. For numerical calculations, where new perturbations are constantly introduced, the shape will become more and more distorted as time evolves. All stable numerical methods also have some artificial diffusion that will smear the profile. We thus have to stabilize (1) such that $\Phi(x,t) \to \Phi_0(x - vt)$ as $t \to \infty$.

To achieve this, we study the PDE

$$\Phi_t + v\Phi_x = \frac{1}{\mu} \left( \varepsilon \Phi_{xx} - (\Phi(1 - \Phi))_x \right). \quad (3)$$

If $\Phi(x,0) = \Phi_0$, where $\Phi_0$ is given by (2), then $\Phi(x,t) = \Phi_0(x - vt)$, i.e. the analytical solution is identical to the solution of (1). What happens now if we perturb the initial data? Rescaling using $\xi = x - vt, \tau = t/\mu$, and by defining $\Psi(\xi,\tau) = 1 - 2\Phi(\xi,\tau)$ then (3) becomes the well known Burgers equation:

$$\frac{1}{2}(\Psi^2)_\xi + \Psi_\tau = \varepsilon \Psi_{\xi\xi}.$$}

Steady state solutions to this equations are $\Psi_0(\xi + \gamma)$ with $\Psi_0(\xi) = 1 - 2\Phi_0(\xi)$ for any shift $\gamma$. We can thus in general only hope for convergence in time to a shifted profile. The evolution of perturbations to the initial data $\Psi_0(\xi) = 1 - 2\Phi_0(\xi)$ of this equation was studied in [8]. By assuming the perturbations $\delta(\xi,\tau)$ to be small, the evolution of $\delta(\xi,\tau)$ can be approximated by a linearized problem

$$\delta_t = \mathcal{L}\delta.$$}

It was shown in [8] that the spectrum of the operator $\mathcal{L}$ consists of an isolated eigenvalue $\lambda = 0$ plus a continuous spectrum $-\infty < \lambda < -c/\varepsilon$ for some $c > 0$. 

4
It can be shown that the eigenvalue 0 corresponds to a shift in \( \Psi \). This means that we can expect small perturbations to \( \Psi_0(\xi) \) to converge to a shifted \( \Psi \), i.e. \( \Psi(\xi, \tau) \rightarrow \Psi_0(\xi + \gamma) \), as \( \tau \rightarrow \infty \). The convergence to the shifted steady state can be expected to be exponential in time on the time scale \( \tau \sim \varepsilon \). The stability of viscous shocks was also studied in [9]. There it was shown that if the perturbation is of zero mass, i.e. if \( \int_{-\infty}^{\infty} \delta(\xi,0)d\xi = 0 \), then there will be no shift, i.e. \( \gamma = 0 \). Since \( v \) is constant, (1) is also a conservation law. When using conservative numerical schemes, the truncation errors will be of zero mass. We thus expect these errors to decrease rapidly in time without shifting the profile.

The time scale on which the perturbations decrease will be \( t \sim \mu \varepsilon \). We assume the velocity \( v = \mathcal{O}(1) \). If \( \varepsilon \mu \ll 1 \) this time scale will be much faster than the time scale related to the advection. For a constant velocity it is reasonable to choose \( \mu = 1 \), since \( \varepsilon \) is assumed to be small.

Remark: We are only interested in divergence free velocity fields, since then the advection can be written in conservative form. For a one dimensional problem the divergence free condition implies that the velocity is constant. It does therefore not make sense to discuss variable velocities in one dimension.

### 2.2 Stabilized advection in two and three dimensions

For divergence free velocities \( \vec{u} \) in several space dimensions we want to stabilize the profile across the interface in the direction normal to the interface \( \Gamma \). The stabilized advection can be expressed as

\[
\Phi_t + \nabla \cdot (\vec{u} \Phi) = \frac{1}{\mu} \nabla \cdot (-\Phi(1 - \Phi)\hat{n} + \varepsilon (\nabla \Phi \cdot \hat{n})\hat{n})
\]  

where \( \hat{n} = \nabla \Phi / |\nabla \Phi| \). In several space dimensions, the divergence free condition on the velocity does not imply a constant velocity. Then also variations in the velocity will distort the shape of \( \Phi \) across the interface. This implies that it might be necessary to make \( \mu \) smaller in order to keep the profile of \( \Phi \) across the interface. How small \( \mu \) has to be chosen might also vary in time. For example, when two interfaces merges \( \mu \) will have to be smaller. To handle this numerically we split the advection and the stabilization into a set of two PDE:s. First one time step of

\[
\Phi_t + \nabla \cdot (\Phi \vec{u}) = 0
\]  

will be solved. The resulting \( \Phi \) is then used as initial condition to

\[
\Phi_t + \nabla \cdot (\Phi(1 - \Phi)\hat{n}) = \varepsilon \nabla \cdot ((\nabla \Phi \cdot \hat{n}) \hat{n}),
\]
which is then solved to steady state. The process of solving (6) to steady state will be referred to as the reinitialization step. By splitting (4) in this way we do not have to determine how small $\mu$ should be. Moreover, from the discussion in Section 2.1 we can expect the time for convergence of (6) to steady state to be of $O(\varepsilon)$. As will be seen later we will choose the discretization parameters to $\Delta t \sim \Delta x \sim \varepsilon$. This implies that only a few time steps, $O(1)$, will be needed to reach the steady state of (6).

Since both the advective step as well as the reinitialization step are formulated as conservation laws, $\int_\Omega \Phi \, dx$ is constant in time in the continuous case. By using conservative numerical methods, this quantity can also be conserved exactly in the discrete approximation of the PDE:s. The conservative properties of the method will be discussed further in Section 4.

2.2.1 Modification of the reinitialization step

In the original formulation in [7] the reinitialization was given by

$$\Phi_\tau + \nabla \cdot (\Phi (1 - \Phi) \hat{n}) = \varepsilon \nabla \cdot (\nabla \Phi).$$

(7)

If $\hat{n} = \nabla \Phi / |\nabla \Phi|$, then $\nabla \Phi = (\nabla \Phi \cdot \hat{n}) \hat{n}$. This implies that the diffusion will result in a flux in the direction of $\hat{n}$ only. This flux will rapidly become balanced by the compressive flux $\Phi (1 - \Phi) \hat{n}$. Hence, we can expect the motion of the interface during the reinitialization to be small. In practice we will however fix $\hat{n}$ such that $\hat{n}(\tau) = \nabla \Phi_{\tau=0} / |\nabla \Phi_{\tau=0}|$ during each reinitialization step. $\hat{n} = \nabla \Phi / |\nabla \Phi|$ will then only hold for $\tau = 0$. As $\tau$ increases the diffusion might result in a small flux in the direction tangential to the interface. This flux will not be balanced by any compression. Because of this, the tangential diffusion might move the interface. This effect will be particularly strong if a large $\tau$ is needed to reach the steady state of (7). To avoid any tangential diffusion, we have replaced (7) by (6).

We will use finite elements to discretize the reinitialization step. For these to be stable (see for example [10]) diffusion is needed in the direction of the compression only. Because of this we will not have to add any artificial diffusion to stabilize the discretization of (6).

2.3 Discretizations using finite elements

In order to be able to easily apply our method to problems with complex geometries as well as to simplify the use of adaptive grids we use a finite element discretization of the PDE:s in space. In time finite differences are used.
For all finite element approximations we need to define finite dimensional function spaces. Throughout this paper we will denote

\[ V_h = \{ f(x) : f(x) \text{ is piecewise linear within } \Omega \text{ and } f(x) = 0 \forall x \in \Lambda \subset \partial \Omega \}. \]

We will not explicitly define \( \Lambda \), but simply assume it to be the part of the boundary where Dirichlet boundary conditions have been given on the corresponding unknown. Vector valued function spaces with \( d \) components will be denoted by

\[ W_h = \{ \vec{f}(x) = [f_1(x), \ldots, f_d(x)]^T : f_i(x) \text{ is piecewise linear within } \Omega \text{ and } f_i(x) = 0 \forall x \in \Lambda_i \subset \partial \Omega, i = 1, \ldots, d \}. \] (8)

The spatial finite element discretization of (5), the advection step, can now be formulated as: Find \( \Phi \in V_h \) such that

\[ \int_\Omega v \Phi_t dx - \int_\Omega \nabla v \cdot (\Phi \vec{u}) dx + \int_{\partial \Omega} v \phi \vec{u} \cdot \hat{n} dS = 0 \quad \forall v \in V_h. \] (9)

If the boundaries are walls with normal \( \hat{\nu} \), \( \vec{u} \cdot \hat{\nu} = 0 \) and the last term vanishes. The spatial discretization of the reinitialization (6) is given by: Find \( \Phi \in V_h \) s.t.

\[ \int_\Omega v \Phi_r dx + \int_\Omega \nabla v \cdot \left( -\vec{f} + \varepsilon (\nabla \Phi \cdot \hat{n}) \hat{n} \right) dx + \int_{\partial \Omega} v \left( \vec{f} - \varepsilon (\nabla \Phi \cdot \hat{n}) \hat{n} \right) \cdot \hat{\nu} dS = 0 \quad \forall v \in V_h. \] (10)

Here, \( \vec{f} = \Phi (1 - \Phi) \hat{n} \). To avoid any flow through the boundaries, the boundary term should be set to zero.

The temporal discretization of the advection equation (9) is discretized using forward Euler. Let \( \Phi^n \approx \Phi(t_n) \). An intermediate \( \Phi^{n+1}_* \) is then calculated. Find \( \Phi^{n+1}_* \in V_h \) such that

\[ \int_\Omega v \frac{\Phi^{n+1}_* - \Phi^n}{dt} dx - \int_\Omega \nabla v \cdot (\Phi^n \vec{a}^n) dx = 0 \quad \forall v \in V_h \] (11)

One should note that if we do not perform any reinitialization, the discretization (11) will be unstable. Together with the reinitialization step we did not observe any instabilities in our numerical experiments. If necessary, (11) can be stabilized by adding streamline diffusion.

To calculate the reinitialization step, the normal of the interface, \( \hat{n}^{n+1}_* \), has to be approximated. This is also done using finite elements. Find \( \hat{n}^{n+1}_* \in W_h \) s.t.

\[ \int_\Omega \vec{v} \cdot \nabla \frac{\Phi^{n+1}_*}{\nabla \Phi^{n+1}_*} dx = \int_\Omega \vec{v} \cdot \hat{n}^{n+1}_* dx \quad \forall \vec{v} \in W_h. \] (12)
Finally we use a second order accurate discretization in time for the reinitialization step. We start by letting \( m = 0 \) and \( \Phi^0_c = \Phi^{n+1}_c \). Then for \( m = 0, \ldots \) we determine \( \Phi^{m+1}_c \in V_h \) such that

\[
\int_\Omega v \frac{\Phi^{m+1}_c - \Phi^m_c}{d\tau} dx - \int_\Omega \left( \frac{\Phi^m_c + \Phi^{m+1}_c}{2} - \Phi^{m+1}_c \Phi^m_c \right) \nabla v \cdot \hat{n} dx + \varepsilon \nabla \left( \frac{\Phi^m_c + \Phi^{m+1}_c}{2} \right) \cdot \hat{n} \left( \nabla v \cdot \hat{n} \right) dx = 0 \quad \forall v \in V_h \quad (13)
\]

The iteration stops when

\[
\frac{\| \Phi^{m+1}_c - \Phi^m_c \|}{d\tau} < \delta \quad (14)
\]

for some small \( \delta \). Following the discussion in Section 2.1 we expect only a few time steps to be needed to fulfill the condition (14). Finally we set \( \Phi^{n+1} = \Phi^{m+1}_c \).

### 2.4 Restrictions on \( \varepsilon \)

There are some restrictions on how \( \varepsilon \) can be chosen. A too small \( \varepsilon \) compared to the gridsize \( h \) will create over or undershoots in the steady state solution of (13). The corresponding one dimensional problem is to find \( \phi^n \) such that

\[
\int_a^b \phi^n (1 - \phi^n) v_x dx = \varepsilon \int_a^b \phi^n_2 v_x dx \quad \forall v \in V_h.
\]

Choose \( v \) such that \( v_x = 1 \) if \( x \in [x_i, x_{i+1}] \) and zero otherwise. Denoting \( h_i = x_{i+1} - x_i \) and \( \phi_i = \phi^n(x_i) \) we get

\[
h_i \left( \frac{\phi_i + \phi_{i+1}}{2} - \frac{\phi_i^2 + \phi_{i+1}^2 + \phi_i \phi_{i+1}}{3} \right) = \varepsilon (\phi_{i+1} - \phi_i).
\]

If \( \phi_i \approx \phi_{i+1} \approx 0 \) we get by linearizing around zero that

\[
\phi_{i+1} \approx \frac{h_i/2 + \varepsilon}{\varepsilon - h_i/2} \phi_i.
\]

This implies that if \( \varepsilon < h_i/2 \), then \( \phi_i \) and \( \phi_{i+1} \) will be of opposite sign, i.e. \( \phi^n \) will be oscillating. By assuming \( \phi_i \approx \phi_{i+1} \approx 1 \) we obtain the same restriction on \( \varepsilon \). Note that on a non uniform grid restrictions on \( \varepsilon \) will be different on different parts of the grid. A small \( \varepsilon \) is typically desired close to \( \phi = 0.5 \) to get a sharp transition. A sharp transition will imply less smearing of density, viscosity and surface tension. As we shall see in Section 4, a small \( \varepsilon \) will also result in better conservation of the area bounded by the 0.5 contour. Far away
from the interface we can allow a bigger \( \varepsilon \). This can be achieved by having an adaptive grid refined close to the interface. In this way we can allow a smaller \( \varepsilon \) close to the interface without introducing oscillations. The adaptive procedure will be discussed further in Section 5.

3 The incompressible Navier-Stokes equations

The incompressible Navier-Stokes equations for two phase flow with surface tension are given by

\[
(\rho\ddot{u})_t + \nabla \cdot (\rho u \ddot{u}) = -\nabla p + \frac{1}{Re} \nabla \cdot \left( \mu \left( \nabla \ddot{u} + (\nabla \ddot{u})^T \right) \right) + \frac{\rho}{Fr^2} \ddot{e}_g + \frac{1}{We} \ddot{F}_{sv} \quad (15)
\]

\[
\nabla \cdot \ddot{u} = 0 \quad (16)
\]

\[
\Phi_t + \nabla \cdot (\Phi \ddot{u}) = 0 \quad (17)
\]

where \( Re = \frac{\rho_{ref} u_{ref} l_{ref}}{\mu_{ref}} \), \( Fr = \frac{u_{ref}}{\sqrt{l_{ref} g}} \) and \( We = \frac{\rho_{ref} u_{ref}^2 l_{ref}}{\sigma} \). \( \rho_{ref}, \mu_{ref}, l_{ref}, u_{ref} \) are constant reference density, viscosity, length and velocity, and \( \sigma \) is the surface tension. We will use a diffuse interface approach to model the force due to surface tension, introduced in [11]:

\[
\ddot{F}_{sv} = \kappa \nabla \Phi = - \left( \nabla \cdot \frac{\nabla \Phi}{|\nabla \Phi|} \right) \nabla \Phi.
\]

The density and viscosity are constant within each fluid. We let them vary smoothly over the interface according to \( \Phi \):

\[
\rho = \rho_1 + (\rho_2 - \rho_1) \cdot \Phi
\]

\[
\mu = \mu_1 + (\mu_2 - \mu_1) \cdot \Phi. \quad (18)
\]

\( \rho_1, \rho_2 \) and \( \mu_1, \mu_2 \) are the dimensionless densities and viscosities of the two fluids respectively.

Assuming all boundaries to be walls we discretize in space using linear finite element approximations:

Find \( \ddot{u} \in W_h, p \in V_h \) such that

\[
\int_{\Omega} (\rho\ddot{u})_t \cdot \dddot{v} dx - \int_{\Omega} (\dddot{u} \cdot \nabla \ddot{v}) \cdot \rho \ddot{u} dx = \int_{\Omega} (\nabla \cdot \ddot{u}) p dx +
\]

\[
- \frac{1}{Re} \int_{\Omega} \mu \sum_i \nabla v_i \cdot (\nabla u_i + \ddot{u}_i) dx + \int_{\Omega} \ddot{v} \cdot \left( \frac{\rho}{Fr^2} \ddot{e}_g + \frac{1}{We} \ddot{F}_{sv} \right) dx \quad \forall \dddot{v} \in W_h \quad (19)
\]
and

$$\int_{\Omega} q(\nabla \cdot \vec{u}) dx = 0 \quad \forall q \in V_h. \quad (20)$$

We use a projection method ([12], [13]) similar to the unconditionally stable method proposed for one fluid incompressible flow by Guermond and Quartapelle [14]. Using this method we do not obtain any severe restrictions in time if the Reynolds number is small. This had been the case if we had used a method taking the viscous term explicitly in time.

Before an intermediate velocity is calculated, approximations of the curvature and the gradient of \( \Phi \) have to be calculated. The approximate gradient, \((\nabla \Phi)^{n+1} \in W_h\), is determined such that

$$\int_{\Omega} (\nabla \Phi)^{n+1} \cdot \vec{v} dx = \int_{\Omega} \nabla (\Phi^{n+1}) \cdot \vec{v} dx \quad \forall \vec{v} \in W_h. \quad (21)$$

The difference between \((\nabla \Phi)^{n+1}\) and \(\nabla (\Phi^{n+1})\) is that \((\nabla \Phi)^{n+1}\) will be piecewise linear and continuous while \(\nabla (\Phi^{n+1})\) is piecewise constant. The mean curvature of the level sets of any \( \Phi \) is given by

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

Assuming a contact angle of 90° i.e. \(\hat{n} \cdot \hat{\nu} = 0\) on \(\partial \Omega\) this can be discretized by finding \(\kappa^{n+1} \in V_h\) such that

$$\int_{\Omega} \kappa^{n+1} \vec{v} dx = \int_{\Omega} \nabla \vec{v} \cdot \frac{\nabla \Phi^{n+1}}{|(\nabla \Phi)^{n+1}|} dx \quad \forall \vec{v} \in V_h. \quad (22)$$

This might however produce spurious oscillations in space in \(\kappa^{n+1}\) with a frequency of about 1/\(h\). These high frequent parts can be damped if we instead calculate a regularized curvature \(\tilde{\kappa}\) by approximating

$$\tilde{\kappa} - \epsilon_2 \Delta \tilde{\kappa} = \kappa = -\nabla \cdot \frac{\nabla \Phi}{|(\nabla \Phi)|}.$$ 

Using standard Fourier methods it is clear that

$$\tilde{\kappa}(\vec{\omega}) = \frac{1}{1 + \epsilon_2 \|\vec{\omega}\|^2} \hat{\kappa}(\vec{\omega})$$

where \(\hat{\kappa}(\vec{\omega})\) and \(\tilde{\kappa}(\vec{\omega})\) are the Fourier coefficients of \(\kappa\) and \(\tilde{\kappa}\) and \(\vec{\omega}\) is a vector of the frequencies in the x, y and z direction. By letting \(\epsilon_2 = h \tilde{\kappa} \approx \hat{\kappa}\) if \(\|\vec{\omega}\| = \mathcal{O}(1)\) and \(\tilde{\kappa} \approx h \hat{\kappa}\) if \(\|\vec{\omega}\| = \mathcal{O}(1/h)\). An approximation of the regularized curvature is: Find \(\tilde{\kappa}^{n+1} \in V_h\) such that

$$\int_{\Omega} \kappa^{n+1} \vec{v} dx = \int_{\Omega} \nabla \vec{v} \cdot \frac{(\nabla \Phi)^{n+1}}{|(\nabla \Phi)^{n+1}|} dx - \epsilon_2 \int_{\Omega} \nabla \vec{v} \cdot \nabla \tilde{\kappa}^{n+1} dx \quad \forall \vec{v} \in V_h. \quad (23)$$
Here we put the boundary condition \( \nabla \kappa^{n+1} \cdot \hat{\nu} = 0 \) on \( \partial \Omega \).

An intermediate velocity \( \vec{u}^{n+1}_* \), which in general is not divergence free, can now be calculated taking the pressure term explicitly.

Find \( \vec{u}^{n+1}_* \in W_h \) such that

\[
\frac{1}{dt} \int_{\Omega} \left( \rho^{n+1} \vec{u}^{n+1}_* - \rho^n \vec{u}^n \right) \cdot \vec{v} dx - \int_{\Omega} (\vec{v} \cdot \nabla \vec{v}) \cdot \rho \vec{u}^{n+1}_* dx = \int_{\Omega} (\nabla \cdot \vec{v}) p^n dx - \int_{\Omega} \mu^{n+1} \sum_i \nabla v_i \cdot (\nabla u_i^{n+1}_* + \vec{u}^n_{x_i}) dx + \int_{\Omega} \vec{v} \cdot \left( \frac{\rho^{n+1}}{Fr^2 e} + \frac{1}{We} \vec{F}^{n+1}_{sv} \right) dx \quad \forall \vec{v} \in W_h
\]

Find \( \vec{u}^{n+1}_* \in W_h \) such that

\[
\frac{1}{dt} \int_{\Omega} \rho^{n+1} \vec{u}^{n+1}_* - \rho^n \vec{u}^n \cdot \vec{v} dx - \int_{\Omega} (\vec{v} \cdot \nabla \vec{v}) \cdot \rho \vec{u}^{n+1}_* dx = \int_{\Omega} (\nabla \cdot \vec{v}) p^{n+1} dx - \int_{\Omega} \mu^{n+1} \sum_i \nabla v_i \cdot (\nabla u_i^{n+1}_* + \vec{u}^n_{x_i}) dx + \int_{\Omega} \vec{v} \cdot \left( \frac{\rho^{n+1}}{Fr^2 e} + \frac{1}{We} \vec{F}^{n+1}_{sv} \right) dx \quad \forall \vec{v} \in W_h.
\]

(24)

We define \( \vec{w}^{n+1} \) similarly, but with the pressure term implicitly. Find \( \vec{w}^{n+1} \in W_h \) such that

\[
\frac{1}{dt} \int_{\Omega} \rho^{n+1} \vec{w}^{n+1} - \rho^n \vec{w}^n \cdot \vec{v} dx - \int_{\Omega} (\vec{v} \cdot \nabla \vec{v}) \cdot \rho \vec{w}^{n+1} dx = \int_{\Omega} (\nabla \cdot \vec{v}) p^{n+1} dx - \int_{\Omega} \mu^{n+1} \sum_i \nabla v_i \cdot (\nabla u_i^{n+1}_* + \vec{u}^n_{x_i}) dx + \int_{\Omega} \vec{v} \cdot \left( \frac{\rho^{n+1}}{Fr^2 e} + \frac{1}{We} \vec{F}^{n+1}_{sv} \right) dx \quad \forall \vec{v} \in W_h.
\]

In order to solve for \( \vec{w}^{n+1} \), the pressure \( p^{n+1} \) is needed. By subtracting (25) from (24) and by requiring \( \vec{w}^{n+1} \) to be divergence free, \( p^{n+1} \) can be solved for. Find \( p^{n+1} \in V_h \) such that

\[
- \frac{1}{dt} \int_{\Omega} q \nabla \cdot \vec{w}^{n+1} dx = \int_{\Omega} \nabla q \cdot \nabla \left( \frac{p^{n+1} - p^n}{\rho^{n+1}} \right) dx \quad \forall q \in V_h.
\]

(26)

Finally, \( \vec{w}^{n+1} \) can be solved for. Find \( \vec{w}^{n+1} \in W_h \) such that

\[
\int_{\Omega} \vec{v} \cdot \frac{\vec{w}^{n+1} - \vec{w}^n}{dt} dx = - \int_{\Omega} \vec{v} \cdot \nabla \left( \frac{p^{n+1} - p^n}{\rho^{n+1}} \right) dx \quad \forall \vec{v} \in W_h.
\]

(27)

We summarize the algorithm for solving both the Navier-Stokes as well as the motion of the interface. In each time step the following steps are executed.

- Calculate \( \Phi^{n+1}_* \) using (11).
- Calculate \( \hat{n}_* \) from \( \Phi^{n+1}_* \) using (12).
- Using \( \Phi^{n+1}_* \) as initial data, solve (13) to steady state. This gives \( \Phi^{n+1} \).
- Calculate \( (\nabla \Phi)^{n+1} \) from \( \Phi^{n+1} \) using (21).
- Calculate \( \kappa^{n+1} \) using (23).
- Calculate \( \tilde{w}^{n+1}_* \) from (24).
• Calculate $p^{n+1}$ from (26).
• Finally calculate $\vec{u}^{n+1}$ using (27).

The implementations were done using the finite element tool Femlego [15] [16].

4 Conservation

We can consider conservation of the method from two perspectives. These are the conservation of mass and the conservation of the area (volume for problems in three dimension) bounded by the 0.5 contour of $\Phi$.

Exact mass conservation of the method for both two and three dimensional problems is easily verified. By taking $v \equiv 1$ in (11) and (13) it is clear that $\int_{\Omega} \Phi^n \, dx = \int_{\Omega} \Phi^{n+1} \, dx$. Since the density is approximated by $\rho^n = \rho_1 + (\rho_2 - \rho_1) \phi^n$, it follows that $\int_{\Omega} \rho^n \, dx = \int_{\Omega} \rho^{n+1} \, dx$. Mass will thus be conserved exactly by the method. In a diffuse interface approach we can interpret $\Phi$ as the concentration of fluid one. The total mass of fluid one, $m_1$, will then also be conserved exactly, since $m_1^{n+1} = \int_{\Omega} \rho_1 \Phi^{n+1} \, dx = \int_{\Omega} \rho_1 \Phi^n \, dx = m_1^n$. Since the total mass of both fluids is conserved, the mass of the second fluid will also remain constant.

Next we study the conservation of the area bounded the 0.5 contour of $\Phi^n$, which we denote by $A_{\Phi^n=0.5}$. Here we will only study the conservation of area for a two dimensional problem, although our derivation easily can be extended to volume conservation of three dimensional problems.

Assume that $\Phi^n$ is a steady state solution of (13) with initial condition $\Phi^n_0$:

$$
\int_{\Omega} \left( \Phi^n - (\Phi^n)^2 \right) \nabla v \cdot \hat{n}_x^n - \varepsilon \nabla v \cdot ((\nabla \Phi^n \cdot \hat{n}_x^n)\hat{n}_x^n) \, dx = 0 \quad \forall v \in V_h. \tag{28}
$$

Here, $\hat{n}_x^n \approx \frac{\nabla \Phi^n}{|\nabla \Phi^n|}$. $\Phi^n$ is thus an approximation to a function $\Phi$ solving

$$
\Phi - \Phi^2 = \varepsilon (\nabla \Phi \cdot \hat{n}). \tag{29}
$$

where, $\hat{n} = \frac{\nabla \Phi^n}{|\nabla \Phi^n|} \approx \hat{n}_x^n$. With $\Phi$ and $\Phi^n$ we associate subspaces of $\Omega$ according to the following definition.

Definition 1. Define the subspaces $\Omega^n_1, \Omega^n_2, \Omega_1, \Omega_2$ and $\Gamma$ of $\Omega$ according to

$$
\Omega^n_1 = \{ x \in \Omega : \Phi^n(x) < 0.5 \} \quad \Omega^n_2 = \Omega \setminus \Omega^n_1
$$

$$
\Omega_1 = \{ x \in \Omega : \Phi(x) < 0.5 \} \quad \Omega_2 = \Omega \setminus \Omega_1
$$
and
\[ \Gamma = \{ x \in \Omega : \Phi(x) = 0.5 \}. \]

Define \( \tilde{\Phi}^n \) by
\[
\tilde{\Phi}^n(x) = \begin{cases} 
1 & \text{if } x \in \Omega^n_2 \\
0 & \text{otherwise.} 
\end{cases}
\]

and correspondingly for \( \tilde{\Phi} \):
\[
\tilde{\Phi}(x) = \begin{cases} 
1 & \text{if } x \in \Omega_2 \\
0 & \text{otherwise.} 
\end{cases}
\]

It follows directly from the definition of \( \tilde{\Phi}^n \) that
\[
A_{\Phi^n=0.5} = \int_{\Omega} \Phi^n dx.
\]

To investigate the conservation of \( A_{\Phi^n=0.5} \) we study the difference of \( A_{\Phi^n=0.5} \) and the exactly conserved quantity \( \int_{\Omega} \Phi^n dx \):

\[
\left| \int_{\Omega} \Phi^n dx - A_{\Phi^n=0.5} \right| = \left| \int_{\Omega} \Phi^n dx - \int_{\Omega} \tilde{\Phi}^n(x) dx \right| = \\
\left| \int_{\Omega} \Phi^n dx - \int_{\Omega} \tilde{\Phi}(x) dx - \left( \int_{\Omega} \Phi dx - \int_{\Omega} \tilde{\Phi} dx \right) + \left( \int_{\Omega} \Phi dx - \int_{\Omega} \tilde{\Phi} dx \right) \right| \leq \\
\| \Phi - \Phi^n \|_{L^1(\Omega)} + \| \tilde{\Phi} - \tilde{\Phi}^n \|_{L^1(\Omega)} + \left| \int_{\Omega} \left( \Phi - \tilde{\Phi} \right) dx \right|. \tag{30}
\]

The size of the first two terms on the right hand side of (30) depend on the order of accuracy of the numerical method. Typically \( \| \Phi - \Phi^n \|_{L^1(\Omega)} \sim C(\Delta x)^p \) where \( p \) increases with the order of accuracy of the approximation. The last term is only related to an exact solution of (29) and will not depend on the numerical method used. Bounds on the last term can be proven under some restrictions on the vector field \( \hat{n} \).

**Definition 2.** Let \( X(x_0, s) \) denote the paths given by
\[
X(x_0, 0) = x_0 \text{ for every } x_0 \in \Gamma \\
\frac{\partial X}{\partial s} = \hat{n}.
\]

Furthermore, let
\[
\Omega_0 = \{ x \in \Omega : \text{There do not exist unique } x_0 \text{ and } s \text{ such that } x = X(x_0, s) \}
\]

An example of one and several paths are given in Figure 1. Typically, we cannot assume \( \Omega_0 = \{0\} \). Often, as in the example in Figure 1, we can however assume \( \int_{\Omega_0} d\Omega = 0 \).
Lemma 1. Assume a function \( \Phi \) of which the 0.5 level set forms a closed smooth curve \( \Gamma \). Let \( L_\Gamma \) be the length of \( \Gamma \) and assume that the magnitude of the curvature is bounded by \( \kappa_\infty \). Also assume that \( \Phi \) fulfills
\[
\Phi - \Phi^2 = \varepsilon (\nabla \Phi \cdot \hat{n})
\]
in \( \Omega \setminus \Omega_0 \) where \( \int_{\Omega_0} \Phi d\Omega = 0 \).

Then it holds that
\[
\left| \int_\Omega (\Phi - \tilde{\Phi}) \, dx \right| \leq L_\Gamma \left( \varepsilon \ln(2) + \kappa_\infty \varepsilon^2 \right).
\] (31)

If the paths \( X(x_0, s) \) are straight non intersecting lines perpendicular to \( \Gamma \) for all \( |s| < b \), then
\[
\left| \int_\Omega (\Phi - \tilde{\Phi}) \, dx \right| \leq L_\Gamma \left( \kappa_\infty \varepsilon^2 + \varepsilon \ln(1 + e^{-b/\varepsilon}) \right).
\] (32)

Note that the second term of the bound given by (32) will decrease exponentially fast to zero as \( \varepsilon \to 0 \).

Proof. Since \( \Phi(x) \) solves (29) it can easily be verified that \( \Phi(X(x_0, s)) = g(s) = \frac{1}{1 + e^{\varepsilon s}} \) holds for any \( x \in \Omega \setminus \Omega_0 \). Define the signed distance function by \( \Psi(x) \), i.e.
\[
\Psi(x) = \begin{cases} 
\min_{x_I \in \Gamma} \|x - x_I\| & \text{if } x \in \Omega_1 \\
-\min_{x_I \in \Gamma} \|x - x_I\| & \text{if } x \in \Omega_2.
\end{cases}
\]

We have that
\[
\int_\Omega \Phi dx - \int_\Omega \tilde{\Phi} dx = \int_{\Omega_1} \Phi dx + \int_{\Omega_2} (\Phi - 1) dx = I_1 + I_2.
\]

Since \( s(x) \) is the distance from \( x \) along a path to some point \( x_0 \) on the interface and \( \Psi(x) \) is the shortest distance to the interface it is clear that \( s(x) \geq \Psi(x) \)
for any \( x \in \Omega \setminus \Omega_0 \). Since \( g(s) \) is monotonically decreasing we have that \( \Phi(x) \leq g(\Psi(x)) \). It follows that

\[
0 < I_1 = \int_{\Omega_1} \Phi \, dx = \int_{\Omega_1 \setminus \Omega_0} \Phi \, dx \leq \int_0^\infty g(l)L(l) \, dl \tag{33}
\]

where \( L(l) \) is the length of the interface corresponding to \( \Psi = l \). Since \( \Gamma \) is smooth there exists an \( a \) such that \( \Psi(x) \) is smooth for all \( x \in \{ x : -a \leq \Psi(x) \leq a \} \). For any \( |l| \leq a \)

\[
L(l) = \int_0^{L_\Gamma} \rho \, dt. \tag{34}
\]

Here \( L_\Gamma = L(0) \) denotes the length of \( \Gamma \). For any \( l \) it holds that

\[
L(l) \leq L_\Gamma (1 + \kappa_\infty |l|). \tag{35}
\]

where \( \kappa_\infty \) is the maximum magnitude of the curvature of \( \Gamma \). (33) and (35) now gives

\[
0 < I_1 \leq L_\Gamma \left( \int_0^\infty g(l) \, dl + \kappa_\infty \int_0^\infty lg(l) \, dl \right).
\]

Similarly one obtains bounds on \( I_2 \):

\[
0 > I_2 \geq -L_\Gamma \left( \int_0^\infty g(l) \, dl + \kappa_\infty \int_0^\infty lg(l) \, dl \right).
\]

Using that

\[
\int_0^\infty g(l) \, dl = \int_0^\infty \frac{1}{1 + e^{l/\varepsilon}} \, dl = \varepsilon \ln(1 + e^{-a/\varepsilon})
\]

and

\[
\int_0^\infty lg(l) \, dl = \int_0^\infty \frac{l}{1 + e^{l/\varepsilon}} \, dl < \int_0^\infty \frac{ldl}{e^{l/\varepsilon}} = \varepsilon^2
\]

we obtain the bound

\[
|I_1 + I_2| < L_\Gamma \left( \varepsilon \ln 2 + \kappa_\infty \varepsilon^2 \right)
\]

If all the paths \( X(x_0, s) \) are straight, non intersecting lines for all \( |s| < b \), then \( \Phi(x) = g(\Psi(x)) \) and cancellation of errors gives a stricter bound:

\[
|I_1 + I_2| < L_\Gamma \left( \kappa_\infty \varepsilon^2 + \varepsilon \ln(1 + e^{-b/\varepsilon}) \right).
\]

\[
\square
\]

We can conclude that

\[
A_{\Phi^n_{0.5}} = A_{\Phi^n_{0.5}} + 2\delta
\]

\[
|\delta| \leq \| \Phi - \Phi^n \|_{L^1(\Omega)} + \| \widetilde{\Phi} - \widetilde{\Phi^n} \|_{L^1(\Omega)} + L_\Gamma \left( \varepsilon \ln 2 + \kappa_\infty \varepsilon^2 \right)
\]

15
for general paths $\Omega$, and for straight paths for $|s| < b$

$$|\delta| \leq \|\Phi - \Phi^n\|_{L^1(\Omega)} + \|\tilde{\Phi} - \tilde{\Phi}^n\|_{L^1(\Omega)} + L_{\Gamma} \left( \kappa_\infty \varepsilon^2 + \varepsilon \ln(1 + e^{-b/\varepsilon}) \right).$$

As will be seen in the section on numerical results, we experimentally obtained variations smaller than $L_{\Gamma} \kappa_\infty \varepsilon^2$. This suggests that the we can assume the paths to be straight and perpendicular to $\Gamma$ sufficiently close to the interface. Paths perpendicular to $\Gamma$ corresponds to that $\hat{n} = \nabla \Phi / |\nabla \Phi|$ on $\Gamma$.

Note that $\delta$ does not explicitly depend on $t$ or $\Delta t$. This implies that the difference in area will not increase or decrease as $t$ becomes larger unless the length or the maximum curvature of the interface is increasing or decreasing in time.

5 Adaptive mesh control

There are two reasons to refine the grid close to the interface. First of all, the position of the interface will be more accurately represented. Secondly, the smearing of the density, viscosity and surface tension across the interface is unphysical. It is therefore desirable to keep the thickness of the transition layer as small as possible. A restriction on how small $\varepsilon$ can be was discussed in Section 2.4. If $\varepsilon$, and hence the thickness of the interface, is too small compared to the grid size there will be oscillations in $\Phi$. In order to have a sharp transition, we need to have a refined grid close to the interface. We are therefore interested in an adaptive algorithm. We design our algorithm such that we can control the mesh with respect to the following parameters:

- $\delta$: Thickness of the interface (proportional to $\varepsilon$).
- $m$: Number of elements across the interface.
- $Ind(x)$: Function indicating how to refine.

Here we will use the indicator function

$$Ind(x) = \frac{1}{\varepsilon} \Phi(x)(1 - \Phi(x))$$

and note that over an interface this is approximately $\nabla \Phi \cdot \hat{n} = |\nabla \Phi|$. Clearly, the indicator function is exponentially small away from the interface and has its maximum when $\Phi = 0.5$. The grid is then defined such that $\int_{K_i} Ind(x) dx \approx Tol$ for each element $K_i$. $Tol$ can be chosen such that there will be $m$ grid points across an interface of thickness $\delta$. First we have to define the thickness of the interface. Assume $\Phi(\vec{x}_1) = 0.05$ and $\Phi(\vec{x}_2) = 0.95$ and that $\vec{x}_1 = X(x_0, s_1)$ and $\vec{x}_2 = X(x_0, s_2)$. We define the thickness of the interface by $\delta = |\vec{x}_2 - \vec{x}_1|$. 

16
Using that the profile is
\[
\Phi(X(x_0, s)) = \frac{1}{1 + e^{s/\varepsilon}}
\]  
(37)
and assuming that the path \( X(x_0, s) \) is straight for \( s_1 \leq s \leq s_2 \) one easily sees that \( \delta = \varepsilon \ln(0.95^2/0.05^2) \approx 6\varepsilon \). The grid size is given by \( a(x) = (\varepsilon \cdot Tol)/(\Phi(1 - \Phi)) \), where \( a(x) \) is the area of the elements. If we assume the elements to be isosceles right triangles, then the length of the hypotenuse is \( hyp(x) = \sqrt{4a(x)} \). Introducing a coordinate system along the interface \( m \), the minimum number of elements across the interface, can be estimated. To do this we again use that the profile of the interface is given by (37).

\[
m = \int_{-3\varepsilon}^{3\varepsilon} 1/hyp(x)dx = \frac{1}{2\sqrt{\varepsilon \cdot Tol}} \int_{-3\varepsilon}^{3\varepsilon} \sqrt{\Phi(1 - \Phi)}dx = \\
= \frac{1}{2\sqrt{\varepsilon \cdot Tol}} \int_{-3\varepsilon}^{3\varepsilon} \sqrt{\frac{e^{-x/\varepsilon}}{(1 + e^{-x/\varepsilon})^2}} dx = \frac{\sqrt{\varepsilon}}{2\sqrt{Tol}} \int_{e^{-3\varepsilon}}^{e^{3\varepsilon}} \frac{1}{\sqrt{t(1 + t)}} dt = \\
= \frac{\sqrt{\varepsilon}}{\sqrt{Tol}} \left( \arctan(\sqrt{e^{3\varepsilon}}) - \arctan(\sqrt{e^{-3\varepsilon}}) \right) = const \frac{\sqrt{\varepsilon}}{\sqrt{Tol}} \approx 1.1 \frac{\sqrt{\varepsilon}}{\sqrt{Tol}}
\]  
(38)

Thus \( Tol = \frac{1.3\varepsilon}{m^2} \) will yield approximately \( m \) grid points over the interface. Correspondingly for three dimensional meshes, one easily sees that \( Tol \sim \varepsilon^2/m^3 \).

In order not to make the grid too coarse away from the interface, we make sure \( h \leq h_{\max} \) for some appropriate \( h_{\max} \). We now have expressions on how \( \varepsilon \) and \( Tol \) should be chosen in terms of \( m \) and \( \delta \):

\[
\varepsilon = \frac{\delta}{\ln(0.95^2/0.05^2)} \approx \delta/6
\]  
(39)

\[
Tol = \frac{0.2\delta}{m^2}
\]  
(40)

As was shown in Section 2.4, we will get oscillations in \( \Phi \) when \( \Phi \approx 0 \) or \( \Phi \approx 1 \) if \( \varepsilon \) is too small compared to the grid size \( h \). Because of this we used a larger \( \varepsilon \) in the coarser parts of the grid such that \( \varepsilon \geq Ch \) was fulfilled everywhere.

Since \( \Phi \) will depend on time, the mesh will have to be adjusted in time. How often remeshing is needed depends on how quickly the interface is moving.

6 Numerical results

The performance and convergence of the method was tested on different problems where the motion was driven by surface tension. The gravitational forces are supposed to be small in our future applications. The gravity was therefore set to zero in all the computations presented here.

17
6.1 Oscillating droplet for small Reynolds number in 2D and 3D

In the first test problem the parameters in the Navier-Stokes were set such that the main forces were expected to be surface tension and viscous forces. The dimensionless parameters were set to \( Re = 1, \ We = 1, \ \rho_1 = 10, \ \rho_2 = 1, \ \mu_1 = 10 \) and \( \mu_2 = 1 \). The initial shape of the interface was constructed by a \( 1 \cdot 0.3 \) rectangle and two half circles of radius 0.15 in 2D. For calculations in 3D a cylinder of radius 0.15 and length 1 with two half spheres of radius 0.15 at the ends were used.

6.1.1 Estimation of order of accuracy with respect to grid size

First 2D computations were performed to check the convergence of the method. The computational domain was \( 2 \cdot 2 \) and was discretized using three different uniform meshes of \( 80 \cdot 80, 116 \cdot 116 \) and \( 160 \cdot 160 \) nodes. The position of the 0.5 contour on the different grids for some time steps are shown in Figure 2. On each grid we used: \( \varepsilon = \Delta x \cdot \sqrt{2} \). \( \Delta t \) was proportional to \( \Delta x \). The order of accuracy was calculated with respect to the average distance between the 0.5 contours of the exact solution \( \Phi \) and a numerical solution \( \Phi^n \), i.e.

\[
E = \int_\Omega \frac{|\Phi^n - \Phi|}{L_\Gamma} dx \approx Ch^p. \tag{41}
\]

\( \Phi^n \) and \( \Phi \) go sharply from zero to one at the 0.5 contour of \( \Phi^n \) and \( \Phi \) as in Definition 1. Since the exact solution of the problem is not known, (41) cannot be used directly to calculate the order of accuracy \( p \). By comparing solutions obtained using three different grid sizes, \( h_1, h_2 = k h_1, h_3 = k^2 h_1 \), the order of accuracy can be estimated from

\[
p \approx \frac{\ln \left( \frac{\int_\Omega |\Phi_{h_2} - \Phi_{h_3}| dx - \int_\Omega |\Phi_{h_1} - \Phi_{h_2}| dx}{\ln k} \right)}{\ln k}.
\]

The variation of the area bounded by the 0.5 contour of \( \Phi \) and the estimated order of accuracy at different times are given in Figures 3(a) and 3(b). The area fluctuations are less than 1% on all grids. They are all also smaller than \( \max_i (\kappa_\infty L_\Gamma \varepsilon^2) \), indicating that the bound (32) is more adequate than (31). The order of accuracy is about two. This was the best we could hope for since we were using linear finite element discretizations. The convergence obtained here was much better than for the fluid flow tests in [7]. We believe that the modification in the reinitialization step is the main reason of this improvement. In time, first order discretizations were used. Since our results show second order of accuracy, the error due to the temporal discretization appears to be neglectable compared to the spatial errors. This justifies the use of only first
order accurate discretizations in time.

Fig. 2. Convergence study of oscillating droplet with Re=1. Results using three different grids, each with $\varepsilon = \sqrt{2} \Delta x$

6.1.2 Estimation of error with respect to $\varepsilon$

On the finest grid with 160-160 nodes, runs were also performed using different $\varepsilon$, see Figure 4. This was done to investigate how the size of $\varepsilon$ effects the
solution. We are interested in the behavior of the error given by

$$E_{\varepsilon} = \int_{\Omega} \frac{|\tilde{\Phi}_0 - \tilde{\Phi}_\varepsilon|}{L_T} dx \tag{42}$$

where the subscript of \( \Phi \) denotes the size of \( \varepsilon \). Several factors contribute to this error. First of all the smearing of density, viscosity and surface tension will result in an error. Secondly, the reinitialization will introduce some additional error. Because of the complexity it is hard to analyze the total effect of all these errors. We can however easily numerically estimate an order of accuracy \( p \) with respect to this error if we assume \( E_{\varepsilon} \approx C \varepsilon^p \). Area conservation and estimated order of accuracy with respect to \( \varepsilon \) are given in Figures 5(a) and 5(b). Again, the variations in area suggest the bound (32) to be accurate. For this specific case it seems as if the error with respect to the average distance between the actual interface and the exact interface is \( O(\varepsilon^p) \) with \( p \geq 2 \).

### 6.1.3 Results using adaptive grids in two and three dimensions

Computations were also done using adaptive grids with \( \varepsilon \) set to 0.018. Since the grid was updated during the calculation the number of grid points varied, but was approximately 8000. An example of the grid is shown in Figure 6. In Figure 7 a comparison between results obtained using the adaptive grid and the uniform grids are given. Results on this adaptive grids compared well with the calculations performed using \( 160 \cdot 160 = 25600 \), i.e. using about three times as many nodes. The total computing time was thus reduced to 30%. The gain obtained by using adaptive grids can be expected to be larger for three dimensional computations.

Using the same parameters in the Navier-Stokes equations, three dimensional calculations were also performed. The initial shape of the interface was produced by a cylinder and two half spheres. Adaptive grids were used with
\( \varepsilon = 0.018 \) and about 50,000 nodes. The volume bounded by the 0.5 isosurface varied with about one percent.
6.2 Oscillating liquid Cobalt droplet in air

Cemented carbides are extremely hard materials used for example in tools for steel cutting. The most common cemented carbide consist of Tungsten Carbide (WC) and Cobalt and is constructed in the following way. A powder of Tungsten Carbide and Cobalt is mixed. The mixture is heated such that the Cobalt becomes liquid. WC has an extremely high melting point and remains solid during the sintering. The liquid Cobalt glues the Tungsten Carbide grains.
Fig. 8. Results from three dimensional calculations of viscous droplet (Re=1).

Together and densifies the material. One believes that the driving force during this process is surface tension. A brief description of the liquid phase sintering as well as properties of liquid Cobalt can be found in [17].

In our second simulation we have set density, viscosity and surface tension so that it corresponds to a droplet of liquid Cobalt in air. The length scale is proportional to the size of the powder grains. The dimensionless parameters become $Re = 1000$, $We = 1.2$, $\rho_1 = 0.0001$, $\rho_2 = 1$, $\mu_1 = 0.04$ and $\mu_2 = 1$. The shape of the initial droplet is a 0.6 · 0.5 rectangle with two half circles with radius 0.25 at the end. Calculations were performed on three different grids to check the convergence. The position of the interface on the different grids are shown in Figure 9. The velocity field obtained on the finest grid are shown in Fig. 10.

The order of accuracy with respect to the average distance between the exact interface and the numerically obtained interface was calculated at $t=0.1$, 0.2, ..., 1. The result is shown in Figure 11(a). The area of the 0.5 contour was also calculated at the same time points and is shown in Figure 11(b). We obtain good area conservation. From Figure 9 it is clear that the shape of the droplet at $t=0$ and $t=0.8$ are very similar. Because of this we can expect the area bounded by the interface at $t=0$ and $t=0.8$ to be very similar. This is verified by our calculations as can be seen in Figure 11(b).

To make adaptive runs for this test case did not turn out to be useful. Because of the high Reynolds number the velocity field is varying rapidly in space. This means that we cannot have a very coarse grid away from the interface, since we need to resolve gradients in velocity properly. In an adaptive mesh control suitable for this problem, the criterion for refining should also depend on gradients of $\vec{u}$. 

23
A three dimensional droplet of liquid Cobalt was also studied. In the three dimensional case velocities vary even more rapidly. Computations could therefore not be done using our simple adaptive approach. Instead we used a uniform mesh of $40 \times 40 \times 40 = 64000$ nodes. $\varepsilon = 0.028$ was used. Because of the symmetry of the problem the size of the computational domain of $2 \times 2 \times 2$ could be reduced to $1 \times 1 \times 1$. As can be seen in Figure 12 the motion of the interface is significantly different compared to the case with lower Reynolds number. The velocity field at a few different times is given in Figure 13. At $t=0.6$ steep
gradients of the velocity across the interface can be observed. Eventually the grid is not fine enough to resolve the variations in the velocity field.

7 Conclusions

In this paper we have continued to develop the conservative level set method introduced in [7]. An important improvement is the modification of the reinitialization step. The numerical results in this paper show much better conver-
Fig. 12. Results obtained by three dimensional calculation for liquid Cobalt droplet in air.

gence compared to the rising bubble in the previous paper. We are convinced that the reason for this is the modification of the reinitialization.

We have also given a theoretical analysis on how the area bounded by the appropriate level set can be expected to be conserved. The theory agrees well with results from our numerical computations.

An adaptive finite element discretization of the method has been proposed and implemented in two and three dimensions. Several numerical experiments have been done to investigate the performance of the method. The results from the calculations showed good conservation and convergence, both with respect to the grid size and $\varepsilon$. The adaptive procedure presented here was only based on the distance to the interface. This turned out to work well for problems with a small Reynolds number. In the numerical experiments of the liquid Cobalt droplet the Reynolds number was higher. Because of high gradients in the velocity the criterion how to refine the grid became to crude. It should however be straightforward to take variations in the velocity into account in the adaptive process.
Fig. 13. The velocity field and the interface ($\Phi = 0.5$) for three dimensional Cobalt droplet.

8 Acknowledgements

The authors wish to thank the members of the Computational Phase Transformation project at KTH (www2.mech.kth.se/phasetransf/index.htm). In particular, Walter Villanueva at KTH Mechanics is acknowledged for valuable help on how to use Femlego. This work was supported by SSF grant A3 02:123.

References


