NTNU

Norwegian University of
Science and Technology

Simulation of two-phase flow with varying surface tension.

Karl Yngve Lervåg

Master of Science in Physics and Mathematics
Submission date: June 2008
Supervisor: Einar Rønquist, MATH
Co-supervisor: Svend Tollak Munkejord, SINTEF Energiforskning AS

Norwegian University of Science and Technology
Department of Mathematical Sciences

Problem Description

The main objective with this work is to study the effects of a variable surface tension along an interface separating two fluids by using an existing computational framework owned by and developed at SINTEF Energy Research for studying two-phase flows.

A nonuniform surfactant concentration on the interface is one source of surface tension gradients. An investigation should be made to find a model that describes the surfactant concentration and how surfactants affect the surface tension.

Assignment given: 15. January 2008
Supervisor: Einar Rønquist, MATH

Samandrag

Denne oppgåva er ei studie av effektane som følgjer av varierande overflatespenning på ei overflate mellom to fluid. Varierande overflatespenning fører til krefter i tangentretninga langs overflata, noko som ofte vert kalla Marangoni-effekten. Desse kreftene er diskutert i detalj, og to testproblem blir studerte for å analysere Marangoni-effekten og verifisere implementasjonen. Den første testen fokuserer på flate grenseflater mellom fluida og ser på stasjonærløysinga for ein lineært varierande overflatespenning, medan den andre testen ser på kreftene på ein to-dimensjonal drope som følgje av varierande overflatespenning. Testane viser at den noverande implementasjonen kan simulere tofase-strøymningar der variasjonen av overflatespenninga er gitt.

Den underliggjande modellen er ein tofase-modell for newtonske fluid med konstant viskositet og tettleik. Tofase-modellen er basert på Navier-Stokes-likningane med ei singular overflatekraft, som saman med forskjellen i eigenskapane til fluida introduserer diskontinuitetar over overflata. Navier-Stokes-likningane blir løyste med ein projeksjonsmetode, og ein kombinasjon av level-set-metoden for å følgje overflata og ghost-fluid-metoden (GFM) for å behandle overflate-diskontinuitetane.

Oppgåva diskuterer også effekten av surfaktantar på overflata. Surfaktantar på ei overflate reduserer den lokale overflatespenninga, og ein ikkje-uniform surfaktantkonsentrasjon fører til varierande overflatespenning og Marangoni-effekten. Ein surfaktantmodell er undersøkt, der ei transportlikning er brukt for å følgje surfaktantkonsentrasjon samt ei tilstandslikning for å kople overflatespenninga med surfaktantkonsentrasjonen.

Abstract

This thesis is a study on the effects of varying surface tension along an interface separating two fluids. Varying surface tension leads to tangential forces along the interface. This is often called the Marangoni effect. These forces are discussed in detail, and two test cases are considered to analyse the Marangoni effect, and to verify the present implementation. The first test studies steady-state two-phase flow where the fluids are separated with plane interfaces and the flow is driven by a linear surface-tension gradient. The second case is an analysis of the initial forces on a two-dimensional droplet due to a linear surface-tension gradient. The tests indicate that the present implementation is capable of simulating two-phase flow with a given surface-tension gradient.

The underlying model is a two-phase flow model for Newtonian fluids with constant viscosity and density. The two-phase model is based on the Navier-Stokes equations coupled with a singular surface force, which together with the difference in fluid properties induces discontinuities across the interface. The Navier-Stokes equations are solved using a projection method, and a combination of the level-set method for capturing the interface and the ghost-fluid method (GFM) for handling the interface discontinuities.

The thesis also discusses the effect of surfactants on an interface. The presence of surfactants reduces the local surface tension, and a non-uniform surfactant distribution results in varying surface tension and the Marangoni effect. A surfactant model is reviewed, where an equation of state couples the surface tension to the surfactant concentration and a transport equation is used to solve the surfactant mass-conservation.

Preface

This thesis will be submitted for the degree of Master of Science and Technology (Sivilingeniør), and it concludes my education at NTNU (Norwegian University of Science and Technology).

I am grateful for having had the opportunity to work with many truly interesting problems, even though it has been a lot of hard work. I have learned that many things that at first seem pretty simple, might actually involve quite a lot of complicated work. And when one problem is solved, two new problems are likely to emerge.

It has been a pleasure to meet regularly with my two supervisors, Professor Einar Rønquist at NTNU and Dr. Svend Tollak Munkejord at SINTEF Energy Research. I would like to thank them both for many interesting discussions and conversations, and for many valuable advices.

I am also thankful for all the extra meetings I had with Dr. Svend Tollak Munkejord, and for all the help I have had with the numerical code and with Tecplot. He also helped keeping my motivation high when the work did not go as smoothly as expected.

Finally, I would like to thank Professor Iver H. Brevik for his helpful insight on some matters regarding the Marangoni force, and Dr. Erik Bjørklund for some details on the jump conditions and the convective term.

Trondheim, June 2008

Karl Yngve Lervåg

Contents

Samandrag iii

Abstract v

Preface vii

List of Figures xi

List of Tables xiii

Nomenclature xv

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Objectives . . . 2
- 1.3 Present contribution . . . 2
- 1.4 Survey of the report . . . 3

2 Fluid dynamics 5
- 2.1 The Navier-Stokes equations . . . 5
- 2.2 The jump conditions across an interface . . . 7
  - 2.2.1 Derivation of the general jump conditions . . . 7
  - 2.2.2 The jump conditions with Marangoni terms . . . 10
- 2.3 Decoupling the pressure from the velocity . . . 11
- 2.4 Summary . . . 13

3 Discretization and numerical methods 15
- 3.1 The projection method . . . 15
- 3.2 The level-set method . . . 16
  - 3.2.1 The level-set function . . . 16
  - 3.2.2 The level-set advection equation . . . 17
- 3.3 Spatial discretization . . . 17
  - 3.3.1 The gradient, divergence and Laplacian operators . . . 18
  - 3.3.2 The convective operator . . . 18
- 3.4 The ghost-fluid method . . . 23
- 3.5 Temporal discretization . . . 25
  - 3.5.1 The SSP RK schemes . . . 26
  - 3.5.2 The time-step restriction . . . 27
- 3.6 Implementation details . . . 28
  - 3.6.1 The Marangoni term . . . 28
  - 3.6.2 Boundary conditions . . . 28
- 3.7 Summary . . . 30

4 Analysis of the Marangoni force 31

4.1 The Marangoni effect 31
4.2 Plane interfaces 32
4.2.1 A horizontal interface 32
4.2.2 A tilted interface 33
4.3 A droplet in a linear temperature field 36
4.3.1 Initial analysis 37
4.3.2 Numerical results 40
4.4 Summary 42

5 Surfactants **45**
5.1 Introduction 45
5.2 A surfactant model 46
5.3 Summary 48

6 Conclusions and recommendations for further work **49**
6.1 Conclusion 49
6.2 Recommendations for further work 49

Bibliography **51**

List of Figures

1.1 Two water droplets in oil 2

2.1 A sketch of a two-phase domain 6
2.2 The domain Ω_ε covering the interface, Γ. 8

3.1 Example of a staggered grid 18
3.2 A comparison of the different discretization methods for the convective term. . . 22
3.3 The split cell (i, j) 24
3.4 Butcher tables for the relevant SSP RK schemes. 26

4.1 Marangoni simulation set-up for a horizontal interface 32
4.2 Flow velocity profiles. 34
4.3 An idealized droplet 37
4.4 Surface-tension forces. 38
4.5 The initial pressure inside the droplet. 41
4.6 The velocity vector-fields close to the droplet. 43

5.1 Sketches of surfactants. 45

List of Tables

3.1 Simulation run times for the different discretization schemes. 23

4.1 Right-hand side vector errors, $E(n_i)$. 35
4.2 Errors in the convective term for the tilted case. 36
4.3 Calculations of the initial force on the droplet. 40

Nomenclature

Latin letters

$\mathbf{1}$	Identity tensor	
$\boldsymbol{a}$	Numerical right-hand side vector	$\mathrm{m/s^2}$
$\mathrm{d}(\boldsymbol{x})$	Distance function	
$\boldsymbol{D}$	Deformation tensor, see equation (2.4)	$1/\mathrm{s}$
D	Arbitrary, bounded domain	
∂D	Boundary of the domain D	
D_s	Diffusion tensor	
D^k	The kth divided difference	
$\boldsymbol{e}_x$	Unit vector in x-direction	
$\boldsymbol{e}_y$	Unit vector in y-direction	
f	Concentration of surfactants	
$\boldsymbol{f}$	Body forces	$\mathrm{N/m^3}$
$\boldsymbol{f}_\mathrm{sfd}$	Surface-force density, see equation (2.9)	$\mathrm{N/m^3}$
$\boldsymbol{f}_\mathrm{s}$	Singular surface force, see equation (2.8)	$\mathrm{N/m^3}$
$\boldsymbol{J}$	Numerical jump tensor, see equation (3.55)	
$m(t)$	An arbitrary material surface element	
$\boldsymbol{n}$	Unit normal vector	
p	Fluid pressure	Pa
Q_j	jth order polynomial	
R	Ideal gas constant	J/mol K
S	Sign function, see equation (3.14)	
$\boldsymbol{T}$	Stress tensor, see equation (2.3)	Pa
T	Temperature	K
$\boldsymbol{t}$	Unit tangent vector	
t	Time	s
$\boldsymbol{u}$	Velocity vector	m/s
$\boldsymbol{u}_I$	Interface velocity	m/s
$\boldsymbol{u}_\mathrm{s}$	Surface component of the velocity	m/s
u	x-component of the velocity	m/s
$\boldsymbol{w}$	Arbitrary vector field	
$\boldsymbol{x}$	Position vector	
$\boldsymbol{x}_I$	A point at the interface	

Greek letters

$\boldsymbol{\delta}$	Delta function	
$\partial\Omega$	Exterior boundary	
ε	An arbitrary small real number	
Γ	The interface separating two fluids	
κ	The local curvature, see equation (3.11)	
λ	Bulk viscosity	$\mathrm{Pa\cdot s}$
μ	Viscosity	$\mathrm{Pa\cdot s}$
ν	Kinematic viscosity	$\mathrm{m^2/s}$

Ω_ε	Domain that coveres the interface
Ω	The flow domain
ω	WENO weights
ϕ	Level-set function, see equation (3.9)
φ	Arbitrary, smooth test function
ρ	Fluid density . . . . . $\mathrm{kg/m^3}$
τ	Pseudo-time

Operators

$[\cdot]$	Difference operator (across the interface), see equation (2.17)
$\boldsymbol{\nabla}_\mathrm{s}$	Surface gradient operator, see equation (2.10)
$\boldsymbol{C}$	Numerical convective operator
$\mathrm{D}/\mathrm{D}t$	The convective derivative (page 8)
D^-	Backward difference, see equation (3.23)
D^+	Forward difference, see equation (3.23)
$\boldsymbol{D}$	Numerical divergence operator
$\boldsymbol{G}$	Numerical gradient operator
$\boldsymbol{L}$	Numerical Laplacian operator
$\boldsymbol{P}$	Orthogonal projection operator, see equation (2.52)

Subscripts and Superscripts

i, j	Spatial indices in the x- and y-direction
$+/-$	Denotes the different fluids
x	Derivative in x-direction

Abbreviations

CFL	Courant-Friedrich-Lewy
ENO	Essential non-oscillatory
GFM	Ghost-fluid method
IIM	Immersed interface method
ODE	Ordinary differential equation
RK	Runge-Kutta
SSP	Strong stability-preserving
WENO	Weighted essential non-oscillatory

1 Introduction

1.1 Background

The mixture of oil and water produced at an oil well often consists of a continuous oil phase containing a large amount of very small water droplets. The amount of water can be as high as 30%, but the quality of the oil is often considered by customers as bad if it contains more than 0.5% water. To remove the water from the oil, the water-containing oil is put in a separation vessel. The most basic separation vessel is the gravity separator. Because water has a higher density than oil, gravity forces the water droplets to sink to the bottom of the vessel. There the water droplets form a dense-packed layer, and eventually they coalesce and form a continuous water-phase which can be extracted. This separation process can be a bottleneck in offshore oil production, both because of the time the oil spends in the separation vessels, and because of the size of the tanks: A gravity separator can be 5 meter in diameter and 20 meter long (Lundgaard, 2005).

The gravity separator can be enhanced by adding electrodes to the sedimentation tanks. By utilizing electric fields, attractive forces can be induced between the droplets and make them coalesce before reaching the bottom of the tank. This process is called electrocoalescence, and it has two main enhancements over the gravity separator:

- The droplets that coalesce before reaching the bottom layer become larger, and therefore fall faster.
- The number of droplets needing to coalesce in the bottom layer is reduced.

Many years of research have been put into the electrocoalescence process, an effort which has resulted in coalescers and more efficient separation vessels. More detailed understanding is needed, though, to create even more efficient coalescers. SINTEF Energy Research has an ongoing project where the water-in-oil sedimentation process is being studied, see e.g. Lundgaard (2005). Part of this project focuses on developing a model for simulating multi-drop behaviour in liquid flows on a macro-scale. The macro-scale model requires a thorough understanding of the interaction between the drops. A second model is therefore developed to study the forces between droplets on a micro-scale. This model is based on Hansen (2005), where a framework for studying detailed droplet interactions in electric fields is presented. The framework was used in Fosse (2006) to study droplet collisions in the presence of an electric field.

The numerical framework created at SINTEF Energy Research for the micro-scale model is able to handle advanced simulations of incompressible, viscous two-phase flows. Interacting, rising bubbles were investigated in Teigen (2007) using the implemented framework, and the ability to model phenomena related to the liquefaction of natural gas was evaluated.

In Hansen (2005), it was suggested to further expand the numerical framework to include a conservation law for surfactants along the interface between the phases. This will allow for the coefficient of surface tension to be a function of surfactant concentration, and will lead to tangential forces dependent on the gradient in the coefficient of surface tension, i.e. Marangoni forces, see e.g. Brennen (2005).

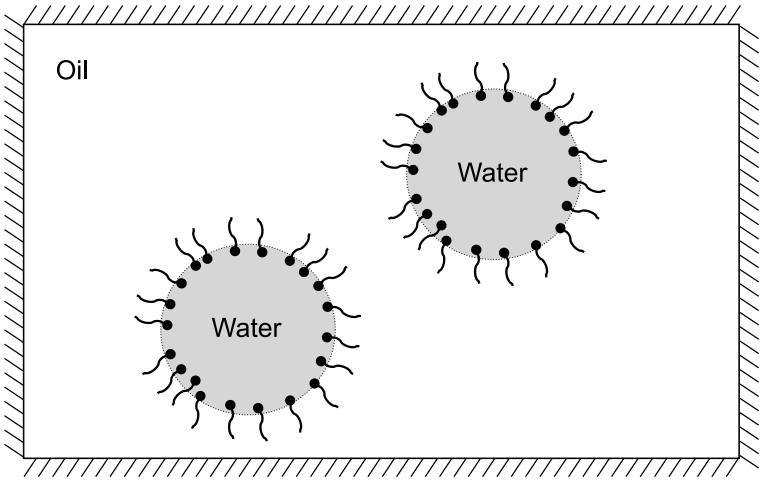


Figure 1.1: Two water droplets in a continuous oil phase with surfactants along the interfaces.

1.2 Objectives

The main goal of this thesis is to begin the work of expanding the numerical framework to include surfactant effects on the interface. This task is quite comprehensive, and requires a deep understanding of the underlying model.

The two-phase model that is currently implemented is based on that described in Hansen (2005). The flow is modelled using the Navier-Stokes equations with an added singular surface force to model the effects of surface tension along the interface. This force and the discontinuities across the interface in fluid properties lead to jump conditions for the velocity and pressure along the interface. The numerical implementation of the model is based on a projection method that decouples the pressure and velocity. Many advanced techniques are involved in the discretization, including the ghost-fluid method for handling the interface discontinuities (Fedkiw, 1999) and the level-set method for capturing and evolving the interface (Osher & Fedkiw, 2003; Sethian, 1999). The governing equations and the numerical methods involved will be reviewed and described in detail.

Figure 1.1 shows an example of a two-phase flow consisting of two water droplets in a continuous oil phase with surfactants attached to the interfaces. The coefficient of surface tension can be related to the surfactant density through an equation of state, and the distribution of surfactants along the interface can be modelled by a transport equation for the surfactants (Xu *et al.*, 2006). In the currently implemented model, the interfacial forces are restricted to the normal direction only. As mentioned above, surfactants on the interface lead to tangential forces as well as normal forces, and the first step towards reaching the main goal is therefore to remove this restriction. The details involved will be described, and the implementation will be tested thoroughly.

The last step is to study and derive a surfactant model that describes the surfactant effects on the interface and the surfactant conservation in the fluid and on the interface.

1.3 Present contribution

The work in this thesis has contributed to the model for studying droplet interactions in several ways. The main contribution is an expansion of the numerical framework to include the Marangoni effect due to varying surface tension. This contribution involves a thorough study of the current numerical

framework in order to implement new functionality. A summary of the important contributions is given below.

- The numerical framework has been expanded to include the Marangoni effect.
- The governing equations and the numerical methods have been reviewed, and the Marangoni effect has been studied and analysed in detail.
- One of the test cases for the Marangoni force required periodic boundaries for the velocity and the level-set function. Periodic boundary conditions have therefore been implemented and tested.
- A study on how surfactants affect the surface tension has been made, and a surfactant model is presented. Ideas for further work on the implementation of the surfactant model are discussed.

1.4 Survey of the report

Chapter 2 An introduction to the fluid dynamics involved in the mathematical model. The equations of motion are reviewed, and special attention is given to the derivation of the jump conditions across an interface.

Chapter 3 The numerical methods involved in the current implementation are reviewed, and some details of the implementation of the Marangoni force are given.

Chapter 4 The Marangoni force is studied, and the implementation of the Marangoni effect is validated with several tests and comparisons with analytic expressions.

Chapter 5 A general model for the effects of surfactants on the interface is reviewed and discussed.

Chapter 6 Presents concluding remarks and suggestions for further work.

2 Fluid dynamics

This chapter will present a mathematical model for an incompressible and Newtonian two-phase flow. In the first section, it will be shown how the Navier-Stokes equations with a singular surface force acting on the interface separating the phases are used to obtain the two-phase flow model.

The numerical model is based on solving the single-phase Navier-Stokes equations in the entire domain, i.e. for both phases. The singular surface force together with the possible discontinuities in the fluid properties, e.g. density and viscosity, induce jumps across the interface in both the pressure and the velocity and their derivatives. These jumps are handled by a set of jump conditions which are derived and discussed in Section 2.2.

In the discretization of the Navier-Stokes equations, a projection method is used to decouple the pressure and velocity. This will result in a Poisson-equation for the pressure coupled with a system of ordinary differential equations (ODEs) for the velocity. The projection method is described for the continuous case in Section 2.3, and is based on the derivation given in Hansen (2005, Chapter 2.2).

Note that all the flows considered in this thesis are two dimensional. This means that the following review of the hydrodynamics will be restricted to two dimensions. Also note that basic details such as the derivation of the Cauchy equation are omitted, as it is assumed that the reader has some familiarity with fluid dynamics. These details can be found in most introductory books on fluid mechanics, e.g. White (2003).

2.1 The Navier-Stokes equations

Under suitable boundary and initial conditions, the Cauchy equation

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t}+\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}\right)=\boldsymbol{\nabla}\cdot\boldsymbol{T}+\rho\boldsymbol{f}, \tag{2.1}$$

and the mass-conservation equation,

$$\frac{\partial \rho}{\partial t}+\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u})=0, \tag{2.2}$$

describe viscous, single-phase flow where temperature effects are neglected. Here ρ is the fluid density, $\boldsymbol{u}$ is the flow velocity, t is time, $\boldsymbol{T}$ is the stress tensor and $\boldsymbol{f}$ denotes body forces. See for example White (2003, Chapter 4) for details and a derivation of these equations. The stress tensor for Newtonian fluids is

$$\boldsymbol{T}=-p\mathbf{1}+2\mu\boldsymbol{D}+\lambda(\operatorname{tr}\boldsymbol{D})\mathbf{1}, \tag{2.3}$$

see for instance Aris (1989, §5) for a good derivation. Here p is the pressure, $\mathbf{1}$ is the identity tensor, μ is the dynamic viscosity, λ is the bulk viscosity, $\operatorname{tr}\boldsymbol{D}$ denotes the trace of the deformation tensor $\boldsymbol{D}$,

$$\boldsymbol{D}=\frac{1}{2}\left(\boldsymbol{\nabla}\boldsymbol{u}+(\boldsymbol{\nabla}\boldsymbol{u})^{T}\right). \tag{2.4}$$

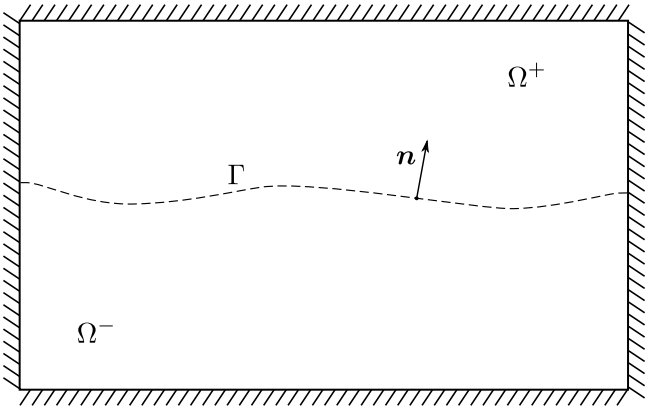


Figure 2.1: A simple two-phase domain where the two phases Ω^+ and Ω^- are separated by the interface Γ. The normal vector, $\boldsymbol{n}$, points from Ω^- to Ω^+.

All fluids considered in this thesis are assumed to be Newtonian, and the viscosities and densities are assumed to be constant. In this case, the stress tensor reduces to

$$\boldsymbol{T} = -p\mathbf{1} + \mu\boldsymbol{\Delta u}, \tag{2.5}$$

and the mass-conservation equation reduces to

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0 \tag{2.6}$$

so that equation (2.1) becomes

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla u} = -\frac{\boldsymbol{\nabla} p}{\rho} + \nu\boldsymbol{\Delta u} + \boldsymbol{f}, \tag{2.7}$$

where $\nu = \mu/\rho$ is the kinematic viscosity. The equations (2.6) and (2.7) are often called the Navier-Stokes equations.

With a simple modification, the same approach can be used to derive the governing equations for two-phase flow. First, let Ω denote the flow domain, let $\partial\Omega$ denote the exterior boundary, let Ω^+ and Ω^- be two immiscible phases, and let Γ denote the interface dividing them, see Figure 2.1. Note that the superscripts $^+$ and $^-$ will be used to denote the different phases throughout the thesis. The extension of the single-phase model to account for two phases can be made by adding a singular surface force term to represent the effects of surface tension between the fluids. The singular surface force is defined as

$$\boldsymbol{f}_{\mathrm{s}}(\boldsymbol{x},t) = \int_\Gamma \boldsymbol{f}_{\mathrm{sfd}}(\boldsymbol{s},t)\boldsymbol{\delta}(\boldsymbol{x}-\boldsymbol{x}_I(\boldsymbol{s}))\,\mathrm{d}\boldsymbol{s}, \tag{2.8}$$

where $\boldsymbol{f}_{\mathrm{sfd}}$ is the surface-force density, $\boldsymbol{\delta}$ is the Dirac delta function and $x_I(\boldsymbol{s})$ is a parametrization of the interface. The position of the interface is assumed to be known. The surface-force density that will be used throughout this thesis is based on the one used in Xu *et al.* (2006),

$$\boldsymbol{f}_{\mathrm{sfd}} = \sigma\kappa\boldsymbol{n} + \boldsymbol{\nabla}_{\mathrm{s}}\sigma, \tag{2.9}$$

where σ is the coefficient of surface tension, κ is the local curvature, $\boldsymbol{n}$ is the normal vector and

$$\boldsymbol{\nabla}_{\mathrm{s}} = (I - \boldsymbol{nn})\cdot\boldsymbol{\nabla} \tag{2.10}$$

is the surface-gradient operator. The notation $\boldsymbol{nn}$ is used for the outer product of vectors, i.e. $\boldsymbol{nn} = \boldsymbol{nn}^T$, and it will be used throughout for the outer products $\boldsymbol{nn}$, $\boldsymbol{nt}$, $\boldsymbol{tn}$ and $\boldsymbol{tt}$, where $\boldsymbol{t}$ is the unit

tangent vector. The terms on the right-hand side of equation (2.9) can be identified as the capillary and the Marangoni force, respectively. For more details concerning the singular surface force, see Hansen (2005, Chapter 2).

The singular surface force (2.8) is added to the Cauchy equation (2.1),

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t}+\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}\right)=\boldsymbol{\nabla}\cdot\boldsymbol{T}+\rho\boldsymbol{f}+\boldsymbol{f}_{\mathrm{s}}. \tag{2.11}$$

The previous assumptions of Newtonian fluids and constant viscosity and density then immediately yield the Navier-Stokes equations with a singular surface force,

$$\frac{\partial \boldsymbol{u}}{\partial t}+\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}=-\frac{\boldsymbol{\nabla}p}{\rho}+\nu\boldsymbol{\Delta}\boldsymbol{u}+\boldsymbol{f}+\frac{\boldsymbol{f}_{\mathrm{s}}}{\rho}, \tag{2.12}$$

and

$$\boldsymbol{\nabla}\cdot\boldsymbol{u}=0. \tag{2.13}$$

Proper boundary conditions are needed to solve the incompressible Navier-Stokes equations. As discussed in Hansen (2005), the issue of well-posed boundary conditions is still an open question. In the two-dimensional case, though, it has been proven that the proper initial and boundary conditions are given by

$$\boldsymbol{u}(\boldsymbol{x},0)=\boldsymbol{u}_0(\boldsymbol{x}) \qquad \forall \boldsymbol{x}\in\Omega, \tag{2.14}$$

$$\boldsymbol{u}(\boldsymbol{x},t)=\boldsymbol{u}_b(\boldsymbol{x},t) \qquad \forall \boldsymbol{x}\in\partial\Omega, \tag{2.15}$$

where $\boldsymbol{u}_0(\boldsymbol{x})$ and $\boldsymbol{u}_b(\boldsymbol{x},t)$ are assumed known. More details on the issue of boundary and initial conditions can be found in Temam (1995, Page 4).

2.2 The jump conditions across an interface

The jump in viscosity and density across the interface can be written as

$$[\mu]=\mu^{+}-\mu^{-}, \tag{2.16}$$

$$[\rho]=\rho^{+}-\rho^{-}, \tag{2.17}$$

where $[\cdot]$ denotes the difference across the interface. This is a convenient notation, and it will be used throughout the following derivation. As stated previously, the singular surface force and the discontinuities in the fluid properties across the interface induce jumps in the pressure and the velocity and their derivatives at the interface. These jumps can be represented by jump conditions for the pressure and the velocity and their derivatives. The derivation of these jump conditions is the topic of this section, and it is mainly based on Hansen (2005, Section 2.4). The section is divided into two parts, where the first part is a derivation of the general jump conditions, and the second part gives the jump conditions for the surface-force density given in equation (2.9).

2.2.1 Derivation of the general jump conditions

In viscous two-phase flow, it is assumed that the velocity is continuous over the interface, see Landau & Lifshitz (1987, Chapter 61). This gives the first jump condition,

$$[\boldsymbol{u}]=0, \tag{2.18}$$

and it forces the tangential derivative of the velocity field to be continuous,

$$[\nabla \boldsymbol{u}] \cdot \boldsymbol{t} = 0. \tag{2.19}$$

If this was not true, the velocity at one side of the interface would develop differently along the interface than the velocity at the other side, which violates the continuity of the velocity. Now, since both fluids are incompressible,

$$[\nabla \cdot \boldsymbol{u}] = \boldsymbol{n} \cdot [\nabla \boldsymbol{u}] \cdot \boldsymbol{n} + \boldsymbol{t} \cdot [\nabla \boldsymbol{u}] \cdot \boldsymbol{t} = 0. \tag{2.20}$$

Combining equations (2.19) and (2.20) yields

$$\boldsymbol{n} \cdot [\nabla \boldsymbol{u}] \cdot \boldsymbol{n} = 0, \tag{2.21}$$

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

To find the remaining jump conditions, consider the domain Ω_ε depicted in Figure 2.2,

$$\Omega_\varepsilon = \{x \in \Omega \mid \min_{x_I \in \Gamma} (x - x_I) \leq \varepsilon\}, \tag{2.22}$$

where $\varepsilon > 0$. Multiply the Cauchy equation for two-phase flow (2.11) with an arbitrary, smooth test function φ with compact support, and integrate over the domain Ω_ε,

$$\int_{\Omega_\varepsilon} \varphi \rho \frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} \,\mathrm{d}\boldsymbol{x} = \int_{\Omega_\varepsilon} \varphi \nabla \cdot \boldsymbol{T} \,\mathrm{d}\boldsymbol{x} + \int_{\Omega_\varepsilon} \varphi \rho \boldsymbol{f} \,\mathrm{d}\boldsymbol{x} + \int_{\Omega_\varepsilon} \varphi \int_\Gamma \boldsymbol{f}_{\mathrm{sfd}} \boldsymbol{\delta}(\boldsymbol{x} - \boldsymbol{x}_I(s)) \,\mathrm{d}s \,\mathrm{d}\boldsymbol{x}, \tag{2.23}$$

where $\mathrm{D}/\mathrm{D}t = \partial/\partial t + \boldsymbol{u} \cdot \nabla$ denotes the convective derivative. Apply the Reynolds transport theorem, see e.g. White (2003, Section 3.2), use that the velocity is continuous across the interface and change the order of integration in the last term to obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega_\varepsilon} \varphi \rho \boldsymbol{u} \,\mathrm{d}\boldsymbol{x} = \int_{\partial\Omega_\varepsilon} \varphi \boldsymbol{T} \cdot \boldsymbol{n} \,\mathrm{d}s - \int_{\Omega_\varepsilon} \boldsymbol{T} \cdot \nabla \varphi \,\mathrm{d}\boldsymbol{x} + \int_{\Omega_\varepsilon} \varphi \rho \boldsymbol{f} \,\mathrm{d}\boldsymbol{x} + \int_\Gamma \varphi \boldsymbol{f}_{\mathrm{sfd}} \,\mathrm{d}s. \tag{2.24}$$

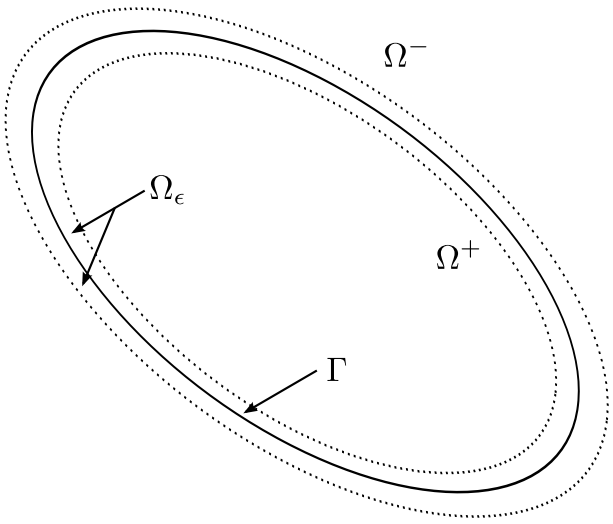


Figure 2.2: The domain Ω_ε covering the interface, Γ.

If ε goes to zero, the left-hand side and the second and third terms on the right-hand side vanish. The first term on the right-hand side becomes the integral of the difference in the stress tensor across the interface, and the resulting equation can be written as

$$0 = \int_\Gamma \varphi \left([\boldsymbol{T}] \cdot \boldsymbol{n} + \boldsymbol{f}_{\text{sfd}} \right) \mathrm{d}s. \tag{2.25}$$

This means that the integrand must be zero, and since the test function is arbitrary,

$$0 = [T] \cdot \boldsymbol{n} + \boldsymbol{f}_{\text{sfd}}. \tag{2.26}$$

Expanding the stress tensor according to equation (2.3) with the usual assumptions of Newtonian and incompressible fluids yields

$$[p]\boldsymbol{n} = [2\mu \boldsymbol{D}] \cdot \boldsymbol{n} + \boldsymbol{f}_{\text{sfd}}. \tag{2.27}$$

The jump condition for the pressure is now obtained by taking the inner product of equation (2.27) with the normal vector $\boldsymbol{n}$, using the identity

$$\boldsymbol{a} \cdot \boldsymbol{B} \cdot \boldsymbol{c} = \boldsymbol{c} \cdot \boldsymbol{B}^T \cdot \boldsymbol{a}, \tag{2.28}$$

together with the equation (2.21), i.e.

$$\begin{aligned}
[p] &= \boldsymbol{n} \cdot [2\mu \boldsymbol{D}] \cdot \boldsymbol{n} + \boldsymbol{f}_{\text{sfd}} \cdot \boldsymbol{n} \\
&= \boldsymbol{n} \cdot \left[\mu \left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T \right) \right] \cdot \boldsymbol{n} + \boldsymbol{f}_{\text{sfd}} \cdot \boldsymbol{n} \\
&= \boldsymbol{n} \cdot [\mu \boldsymbol{\nabla u}] \cdot \boldsymbol{n} + \boldsymbol{n} \cdot [\mu (\boldsymbol{\nabla u})^T] \cdot \boldsymbol{n} + \boldsymbol{f}_{\text{sfd}} \cdot \boldsymbol{n} \\
&= 2[\mu] \boldsymbol{n} \cdot \boldsymbol{\nabla u} \cdot \boldsymbol{n} + \boldsymbol{f}_{\text{sfd}} \cdot \boldsymbol{n}.
\end{aligned} \tag{2.29}$$

To find the jump in the viscous stresses, $[\mu \boldsymbol{\nabla u}]$, it is first rewritten as

$$\begin{aligned}
[\mu \boldsymbol{\nabla u}] = {} & [\boldsymbol{n} \cdot \mu \boldsymbol{\nabla u} \cdot \boldsymbol{n}] \boldsymbol{nn} + [\boldsymbol{t} \cdot \mu \boldsymbol{\nabla u} \cdot \boldsymbol{t}] \boldsymbol{tt} \\
& + [\boldsymbol{n} \cdot \mu \boldsymbol{\nabla u} \cdot \boldsymbol{t}] \boldsymbol{nt} + [\boldsymbol{t} \cdot \mu \boldsymbol{\nabla u} \cdot \boldsymbol{n}] \boldsymbol{tn}.
\end{aligned} \tag{2.30}$$

Equations (2.19) and (2.21) can be used to show that

$$[\boldsymbol{n} \cdot \mu \boldsymbol{\nabla u} \cdot \boldsymbol{n}] = [\mu] \boldsymbol{n} \cdot \boldsymbol{\nabla u} \cdot \boldsymbol{n}, \tag{2.31}$$

$$[\boldsymbol{t} \cdot \mu \boldsymbol{\nabla u} \cdot \boldsymbol{t}] = [\mu] \boldsymbol{t} \cdot \boldsymbol{\nabla u} \cdot \boldsymbol{t}, \tag{2.32}$$

$$[\boldsymbol{n} \cdot \mu \boldsymbol{\nabla u} \cdot \boldsymbol{t}] = [\mu] \boldsymbol{n} \cdot \boldsymbol{\nabla u} \cdot \boldsymbol{t}. \tag{2.33}$$

The simplification of the fourth term from equation (2.30), i.e. $[\boldsymbol{t} \cdot \mu \boldsymbol{\nabla u} \cdot \boldsymbol{n}]$, can be found by taking the inner product of equation (2.27) with $\boldsymbol{t}$,

$$\begin{aligned}
0 &= \boldsymbol{t} \cdot [2\mu \boldsymbol{D}] \cdot \boldsymbol{n} + \boldsymbol{f}_{\text{sfd}} \cdot \boldsymbol{t} \\
&= \boldsymbol{t} \cdot \left[\mu \left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T \right) \right] \cdot \boldsymbol{n} + \boldsymbol{f}_{\text{sfd}} \cdot \boldsymbol{t} \\
&= \boldsymbol{t} \cdot [\mu \boldsymbol{\nabla u}] \cdot \boldsymbol{n} + \boldsymbol{t} \cdot [\mu (\boldsymbol{\nabla u})^T] \cdot \boldsymbol{n} + \boldsymbol{f}_{\text{sfd}} \cdot \boldsymbol{t} \\
&= \boldsymbol{t} \cdot [\mu \boldsymbol{\nabla u}] \cdot \boldsymbol{n} + \boldsymbol{n} \cdot [\mu \boldsymbol{\nabla u}] \cdot \boldsymbol{t} + \boldsymbol{f}_{\text{sfd}} \cdot \boldsymbol{t},
\end{aligned}$$

where the identity from equation (2.28) was used in the last step. This results in

$$[\boldsymbol{t} \cdot \mu \boldsymbol{\nabla u} \cdot \boldsymbol{n}] = -[\mu] \boldsymbol{n} \cdot \boldsymbol{\nabla u} \cdot \boldsymbol{t} - \boldsymbol{f}_{\text{sfd}} \cdot \boldsymbol{t}, \tag{2.34}$$

and the jump in the viscous stress tensor is reduced to

$$\begin{aligned}[\mu\boldsymbol{\nabla u}] = [\mu]\Big(&(\boldsymbol{n}\cdot\boldsymbol{\nabla u}\cdot\boldsymbol{n})\boldsymbol{nn} + (\boldsymbol{n}\cdot\boldsymbol{\nabla u}\cdot\boldsymbol{t})\boldsymbol{nt} \\ &- (\boldsymbol{n}\cdot\boldsymbol{\nabla u}\cdot\boldsymbol{t})\boldsymbol{tn} + (\boldsymbol{t}\cdot\boldsymbol{\nabla u}\cdot\boldsymbol{t})\boldsymbol{tt}\Big) - \big(\boldsymbol{t}\cdot\boldsymbol{f}_{\text{sfd}}\big)\boldsymbol{tn}.\end{aligned} \tag{2.35}$$

The last jump condition is the jump in the gradient of the pressure, $[\boldsymbol{\nabla} p]$. This is a little bit more involved, though a derivation for the simplified case with uniform densities and viscosities and a normal surface tension force can be found in Hansen (2005). In this thesis, the jump is set to

$$[\boldsymbol{\nabla} p] = 0. \tag{2.36}$$

according to Kang *et al.* (2000, Page 340).

For more details on the derivation of these jump conditions, see Hansen (2005). The jump conditions for the generalized surface force density $\boldsymbol{f}_{\text{sfd}}$ are now given in equations (2.18), (2.29), (2.35) and (2.36), repeated here for convenience:

$$\begin{aligned}[\boldsymbol{u}] &= 0, \\ [p] &= 2[\mu]\boldsymbol{n}\cdot\boldsymbol{\nabla u}\cdot\boldsymbol{n} + \boldsymbol{f}_{\text{sfd}}\cdot\boldsymbol{n}, \\ [\mu\boldsymbol{\nabla u}] &= [\mu]\Big((\boldsymbol{n}\cdot\boldsymbol{\nabla u}\cdot\boldsymbol{n})\boldsymbol{nn} + (\boldsymbol{n}\cdot\boldsymbol{\nabla u}\cdot\boldsymbol{t})\boldsymbol{nt} \\ &\qquad - (\boldsymbol{n}\cdot\boldsymbol{\nabla u}\cdot\boldsymbol{t})\boldsymbol{tn} + (\boldsymbol{t}\cdot\boldsymbol{\nabla u}\cdot\boldsymbol{t})\boldsymbol{tt}\Big) - \big(\boldsymbol{t}\cdot\boldsymbol{f}_{\text{sfd}}\big)\boldsymbol{tn}, \\ [\boldsymbol{\nabla} p] &= 0.\end{aligned}$$

A derivation of the jump conditions in three dimensions can be found in Kang *et al.* (2000), although that derivation is restricted to the case where $\boldsymbol{f}_{\text{sfd}} = \sigma\kappa\boldsymbol{n}$. It has been verified that the approach used here is equivalent to the approach in Kang *et al.* (2000) for two dimensions when considering $\boldsymbol{f}_{\text{sfd}} = \sigma\kappa\boldsymbol{n}$.

2.2.2 The jump conditions with Marangoni terms

By use of the expression for the surface force density given in equation (2.9), i.e.

$$\boldsymbol{f}_{\text{sfd}} = \sigma\kappa\boldsymbol{n} + \boldsymbol{\nabla}_{\text{s}}\sigma,$$

the set of jump conditions becomes

$$[\boldsymbol{u}] = 0, \tag{2.37}$$

$$[p] = 2[\mu]\boldsymbol{n}\cdot\boldsymbol{\nabla u}\cdot\boldsymbol{n} + \sigma\kappa, \tag{2.38}$$

$$\begin{aligned}[\mu\boldsymbol{\nabla u}] = [\mu]\Big(&(\boldsymbol{n}\cdot\boldsymbol{\nabla u}\cdot\boldsymbol{n})\boldsymbol{nn} + (\boldsymbol{n}\cdot\boldsymbol{\nabla u}\cdot\boldsymbol{t})\boldsymbol{nt} \\ &- (\boldsymbol{n}\cdot\boldsymbol{\nabla u}\cdot\boldsymbol{t})\boldsymbol{tn} + (\boldsymbol{t}\cdot\boldsymbol{\nabla u}\cdot\boldsymbol{t})\boldsymbol{tt}\Big) - \big(\boldsymbol{t}\cdot\boldsymbol{\nabla}_{\text{s}}\sigma\big)\boldsymbol{tn},\end{aligned} \tag{2.39}$$

$$[\boldsymbol{\nabla} p] = 0. \tag{2.40}$$

The current implementation of the two-phase model is restricted to simulate flows where the surface tension is constant. In these cases, the Marangoni term in the jump condition (2.39) vanishes, i.e.

$$\big(\boldsymbol{t}\cdot\boldsymbol{\nabla}_{\text{s}}\sigma\big)\boldsymbol{tn} = 0. \tag{2.41}$$

With the addition of surfactant effects to the model, this restriction is no longer valid. Surfactants on the interface affect the surface tension, and the surface tension can be written as a function of surfactant concentration. Removing this restriction is therefore a vital step in generalizing the model to include surfactant effects. Implementation details regarding the addition of the Marangoni term will be given in the next chapter. The implementation will be based on the expression for the jump condition (2.39), which is therefore given in its expanded form below. First, the jump condition is written in Cartesian tensor notation,

$$\begin{aligned}[\mu\boldsymbol{\nabla u}]_{ij} = [\mu]\Bigg(&n_k\frac{\partial u_k}{\partial x_l}n_l n_i n_j + n_k\frac{\partial u_k}{\partial x_l}t_l n_i t_j \\ &- n_k\frac{\partial u_k}{\partial x_l}t_l t_i n_j + t_k\frac{\partial u_k}{\partial x_l}t_l t_i t_j\Bigg) - t_k\frac{\partial \sigma}{\partial x_k}t_i n_j,\end{aligned} \tag{2.42}$$

where x_i, u_i, t_i and n_i denote the ith component of the position vector, the velocity vector, the tangent vector and the normal vector, respectively, and the summation convention is used, i.e. $a_k B_{kj} = \sum_k a_k B_{kj}$. The expansion is then found to be

$$\begin{aligned}[\mu\boldsymbol{\nabla u}]_{11} = [\mu]\Bigg(&(n_1^4+n_2^4)\frac{\partial u}{\partial x} + (n_1^3 n_2 - n_1 n_2^3)\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) + 2n_1^2 n_2^2\frac{\partial v}{\partial y}\Bigg) \\ &- n_1 n_2^2\frac{\partial \sigma}{\partial x} + n_1^2 n_2\frac{\partial \sigma}{\partial y},\end{aligned} \tag{2.43}$$

$$\begin{aligned}[\mu\boldsymbol{\nabla u}]_{12} = [\mu]\Bigg(&2n_1 n_2^3\left(\frac{\partial v}{\partial y}-\frac{\partial u}{\partial x}\right) + (2n_1^2 n_2^2 + n_1^2)\frac{\partial u}{\partial y} + (2n_1^2 n_2^2 - n_2^2)\frac{\partial v}{\partial x}\Bigg) \\ &- n_2^3\frac{\partial \sigma}{\partial x} + n_1 n_2^2\frac{\partial \sigma}{\partial y},\end{aligned} \tag{2.44}$$

$$\begin{aligned}[\mu\boldsymbol{\nabla u}]_{21} = [\mu]\Bigg(&2n_1^3 n_2\left(\frac{\partial u}{\partial x}-\frac{\partial v}{\partial y}\right) + (2n_1^2 n_2^2 - n_1^2)\frac{\partial u}{\partial y} + (2n_1^2 n_2^2 + n_2^2)\frac{\partial v}{\partial x}\Bigg) \\ &+ n_1^2 n_2\frac{\partial \sigma}{\partial x} - n_1^3\frac{\partial \sigma}{\partial y},\end{aligned} \tag{2.45}$$

$$\begin{aligned}[\mu\boldsymbol{\nabla u}]_{22} = [\mu]\Bigg(&2n_1^2 n_2^2\frac{\partial u}{\partial x} - (n_1^3 n_2 - n_1 n_2^3)\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) + (n_1^4+n_2^4)\frac{\partial v}{\partial y}\Bigg) \\ &+ n_1 n_2^2\frac{\partial \sigma}{\partial x} - n_1^2 n_2\frac{\partial \sigma}{\partial y},\end{aligned} \tag{2.46}$$

where the tangent components were removed using the identity $\boldsymbol{t}\cdot\boldsymbol{n} = 0$, resulting in

$$\boldsymbol{t} = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} n_2 \\ -n_1 \end{bmatrix}. \tag{2.47}$$

2.3 Decoupling the pressure from the velocity

A projection method is used to numerically solve the Navier-Stokes equations. It is based on the *Helmholtz-Hodge theorem*, which is stated here without proof. Details and proof of the theorem can be found in Chorin & Marsden (2000). A variant of the theorem and a proof can also be found in Aris (1989, §3.44).

> *Any arbitrary vector field* $\boldsymbol{w}$ *can be uniquely decomposed in a bounded domain D such that*
>
> $$\boldsymbol{w} = \boldsymbol{u} + \boldsymbol{\nabla} p, \tag{2.48}$$
> $$\boldsymbol{u}\cdot\boldsymbol{n} = 0 \quad \textit{on } \partial D, \tag{2.49}$$

where p is a scalar function, $\boldsymbol{n}$ is the normal vector pointing out of D and $\boldsymbol{u}$ is any divergence-free vector field, i.e.

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0 \quad \textit{in } D. \tag{2.50}$$

The theorem proves the existence and uniqueness of an orthogonal projection operator $\boldsymbol{P}$, mapping arbitrary vector fields into divergence-free vector fields:

$$\boldsymbol{u} = \boldsymbol{P}\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{\nabla} p. \tag{2.51}$$

By the definition of orthogonal projectors, see for example Saad (2003),

$$\boldsymbol{P}\boldsymbol{u} = \boldsymbol{u}, \tag{2.52}$$

which immediately gives

$$\boldsymbol{P}(\boldsymbol{\nabla} p) = 0. \tag{2.53}$$

It remains to identify the projection operator, which is done by using the orthogonality between the projection operator and the divergence operator, i.e.

$$\boldsymbol{\nabla} \cdot \boldsymbol{P}\boldsymbol{w} = 0. \tag{2.54}$$

Applying the divergence operator to equation (2.51) yields a Poisson equation for p,

$$\begin{gathered} \boldsymbol{\nabla} \cdot \boldsymbol{P}\boldsymbol{w} = \boldsymbol{\nabla} \cdot \boldsymbol{u} = \boldsymbol{\nabla} \cdot \boldsymbol{w} - \boldsymbol{\nabla}^2 p = 0 \\ \Downarrow \\ \boldsymbol{\nabla}^2 p = \boldsymbol{\nabla} \cdot \boldsymbol{w}, \end{gathered} \tag{2.55}$$

and using that $\boldsymbol{u} \cdot \boldsymbol{n} = 0$ on ∂D gives a Neumann boundary condition for equation (2.55),

$$\boldsymbol{n} \cdot \boldsymbol{\nabla} p = \boldsymbol{n} \cdot \boldsymbol{w}. \tag{2.56}$$

These results can be used to decouple the pressure from the incompressible Navier-Stokes equations. First, assume that $\boldsymbol{u}$ is smooth enough, i.e. its first derivative is continuous, and rewrite equation (2.7) to get

$$\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{w} - \frac{\boldsymbol{\nabla} p}{\rho}, \tag{2.57}$$

where

$$\boldsymbol{w} = -(\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{u} + \nu \boldsymbol{\nabla}^2 \boldsymbol{u} + \boldsymbol{f}. \tag{2.58}$$

Now $\partial \boldsymbol{u}/\partial t$ is recognized as the divergence-free vector and $\boldsymbol{w}$ as the arbitrary vector. The pressure can be obtained from the Poisson equation,

$$\frac{\boldsymbol{\Delta} p}{\rho} = \boldsymbol{\nabla} \cdot \boldsymbol{w}, \tag{2.59}$$

with Neumann boundary conditions as for equation (2.55),

$$\boldsymbol{n} \cdot \left(\frac{\boldsymbol{\nabla} p}{\rho} \right) = \boldsymbol{n} \cdot \boldsymbol{w} \quad \text{on } \partial\Omega. \tag{2.60}$$

2.4 Summary

In this chapter, a model for simulating two-phase fluid flow has been presented. The governing equations are the Navier-Stokes equations with a singular surface force given in equations (2.12) and (2.13). The initial and boundary conditions have been discussed, and the jump conditions across the interface were derived. The jump conditions for a surface-tension force that includes both capillary and Marangoni forces are given in equations (2.37) to (2.40). The flow equations were manipulated to decouple the pressure from the velocity, resulting in a new set of equations given in equations (2.57) to (2.59).

3 Discretization and numerical methods

There are many details involved in the numerical solution of the two-phase flow model, some of which will be discussed in this chapter. The reader is referred to Burden & Faires (2004) for basic numerical analysis and results.

The chapter starts with a review of a semi-discretization of the flow equations based on the Helmholtz-Hodge decomposition. It continues with an introduction to the interface-capturing method, followed by a section devoted to the discretization of the spatial operators. An example is given to illustrate the need for using an upwind scheme in the discretization of the convective term. Next, the ghost-fluid method which is used to handle the interfacial discontinuities is discussed. Then the temporal discretization is reviewed, and the chapter concludes with some comments on the implementation details.

3.1 The projection method

There exists a large number of methods for solving the Navier-Stokes equations numerically. In Langtangen *et al.* (2002), an overview of the most common numerical solution strategies for the incompressible Navier-Stokes equations are presented. One can divide the numerical solution of the incompressible Navier-Stokes equations in two main parts: An algorithm for approximating the temporal evolution of the fluid flow and a set of discrete approximations of the continuous differential operators that build up the governing equations.

In this numerical framework, the discretization of the Navier-Stokes equations (2.7) and (2.6) is based on the Helmholtz-Hodge decomposition shown in Section 2.3. First, the Navier-Stokes equations are written on approximate form using numerical spatial approximations for the different terms, i.e.

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{C}(\boldsymbol{u},\boldsymbol{u}) &= -\boldsymbol{G}p/\rho + \nu \boldsymbol{L}\boldsymbol{u} + \frac{\boldsymbol{f}}{\rho}, \\ \boldsymbol{D}\boldsymbol{u} &= 0, \end{aligned} \tag{3.1}$$

where $\boldsymbol{G}, \boldsymbol{C}, \boldsymbol{D}$ and $\boldsymbol{L}$ denote the approximate gradient, convective, divergence and Laplacian operators, respectively. Then the semi-discretized Navier-Stokes equations (3.1) are decoupled in the same manner as the continuous equations were in Section 2.3, resulting in

$$\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{a}^n - \frac{\boldsymbol{G}p^n}{\rho}, \tag{3.2}$$

and

$$\boldsymbol{a}^n = -\boldsymbol{C}(\boldsymbol{u}^n,\boldsymbol{u}^n) + \nu \boldsymbol{L}\boldsymbol{u}^n + \boldsymbol{f}^n/\rho, \tag{3.3}$$

where $\boldsymbol{a}$ is hereafter called the right-hand side vector, and $\boldsymbol{u}^n$, $\boldsymbol{a}^n$ and p^n are the velocity, the right-hand side vector and the pressure evaluated at time $t = t_n$. The Poisson equation becomes

$$\boldsymbol{D}\left(\frac{\boldsymbol{G}p^n}{\rho}\right) = \boldsymbol{D}\boldsymbol{a}^n, \tag{3.4}$$

with Neumann boundary conditions on $\partial\Omega$:

$$\boldsymbol{n} \cdot \frac{\boldsymbol{G}p^n}{\rho} = \boldsymbol{n} \cdot \boldsymbol{a}^n. \tag{3.5}$$

The projection scheme can be summarized with the following simple algorithm for updating the pressure and velocity for each time step:

1. Calculate the right-hand side vector with equation (3.3).
2. Solve the Poisson equation (3.4) to update the pressure.
3. Advance the system of ODEs (3.2) one time step to find $\boldsymbol{u}^{n+1}$.

The Poisson equation in step 2 is mainly solved using the conjugate gradient method with incomplete Cholesky factorization (see e.g. Saad (2003)), while the system of ODEs in step 3 can be advanced using a Runge-Kutta method.

3.2 The level-set method

The temporal evolution of the Navier-Stokes equations assumes that the location of the interface is known. A method which tracks or captures the position of the interface between time steps is therefore essential. There exist several methods for this, many of which are discussed in Hansen (2005, Section 5.4). The level-set method is chosen because of its relative ease of implementation and for its straightforward handling of topological changes. The following derivation of the level-set method is based on Osher & Fedkiw (2003). The interested reader is referred to the book for more details.

3.2.1 The level-set function

In the level-set method, the interface is represented implicitly as the isocontour of some smooth, continuous function, ϕ. ϕ is defined in the whole domain Ω such that

$$\Gamma = \{\boldsymbol{x} \,|\, \phi(\boldsymbol{x}) = 0\}, \tag{3.6}$$
$$\Omega^+ = \{\boldsymbol{x} \,|\, \phi(\boldsymbol{x}) > 0\}, \tag{3.7}$$
$$\Omega^- = \{\boldsymbol{x} \,|\, \phi(\boldsymbol{x}) < 0\}. \tag{3.8}$$

The signed distance function is used in this work as the level-set function, i.e.

$$\phi(\boldsymbol{x}) = \begin{cases} \mathrm{d}(\boldsymbol{x}) \,|\, \boldsymbol{x} \in \Omega^+ \\ -\mathrm{d}(\boldsymbol{x}) \,|\, \boldsymbol{x} \in \Omega^- \end{cases}, \tag{3.9}$$

where $\mathrm{d}(\boldsymbol{x})$ is the distance function, i.e. the minimum distance from the point $\boldsymbol{x}$ to the interface Γ. The signed distance function has the property that $|\nabla\phi| = 1$, and it is smooth and well behaved. This makes it a good choice for the level-set function.

The level-set function is defined such that the normal to the interface pointing from Ω^- to Ω^+ can be calculated as

$$\boldsymbol{n} = \frac{\nabla\phi}{|\nabla\phi|}. \tag{3.10}$$

This makes it easy to find the local curvature,

$$\kappa = -\nabla \cdot \boldsymbol{n} = -\frac{\nabla^2 \phi}{|\nabla \phi|}. \tag{3.11}$$

3.2.2 The level-set advection equation

The evolution of the interface is captured by solving an advection equation of the form

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u}_I \cdot \nabla \phi = 0, \quad \text{in } \Omega, \tag{3.12}$$

where the left-hand side is the convective derivative of the level-set function and $\boldsymbol{u}_I$ is the interface velocity. Since the fluid velocity is continuous across the interface, the interface velocity is equal to the fluid velocity at the interface, $u_I(\boldsymbol{x}_I) = \boldsymbol{u}(\boldsymbol{x}_I)$, where $\boldsymbol{x}_I$ is an interfacial point. For $\boldsymbol{x} \neq \boldsymbol{x}_I$, the velocity $\boldsymbol{u}_I(\boldsymbol{x})$ is found by extrapolation,

$$\frac{\partial \boldsymbol{u}_I}{\partial \tau} + S(\phi_0)\boldsymbol{n} \cdot \nabla \boldsymbol{u}_I = 0, \tag{3.13}$$

where τ is pseudo time and $S(\phi)$ is a smeared sign function,

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + (2\Delta x)^2}}. \tag{3.14}$$

which is based on Sussman *et al.* (1994). One could of course simply use the fluid velocity, $\boldsymbol{u}_I(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x})$, but it is shown that the velocity extrapolation helps to preserve the signed distance-function, see Hansen (2005, Section 5.4.1) and Zhao *et al.* (1996).

When the level-set equation (3.12) is solved numerically, the signed distance-function property will inevitable be lost with time due to numerical errors and smearing. To maintain a signed distance function, the level-set function is reinitialized at regular intervals using an equation proposed by Sussman *et al.* (1994),

$$\frac{\partial \phi}{\partial \tau} + S(\phi)\left(|\nabla \phi| - 1\right) = 0. \tag{3.15}$$

Note that both the reinitialization equation (3.15) and the velocity extrapolation equation (3.13) are solved in pseudo-time, τ. They are both hyperbolic equations, so only a small number of time steps are needed to extend the signed distance function property or the normal velocity field to a narrow region around the interface.

Due to numerical errors in the advection of the level-set function, in addition to numerical smearing in the reinitialization step, the level-set method does not inherently conserve mass. The mass loss is noticeable in regions with high curvature or low grid resolution.

3.3 Spatial discretization

In this section, the spatial numerical operators will be discussed. The section is divided into two parts. The first part gives the discretization of the ordinary spatial operators, e.g. the gradient and the Laplacian, while the second part is a derivation of the weighted ENO (WENO) scheme used to discretize the convective operator.

Note that the spatial discretization utilizes a staggered grid following Osher & Fedkiw (2003), where scalar values are stored in cell centres and vector values are stored on cell boundaries as depicted in Figure 3.1.

3.3.1 The gradient, divergence and Laplacian operators

For the rest of this chapter, let φ be a smooth test-function, let $\boldsymbol{w} = (w^1, w^2)$ be an arbitrary, smooth vector field and let $\boldsymbol{e}_x$ and $\boldsymbol{e}_y$ be the unit vectors in the x- and y-direction. The numerical gradient, divergence and Laplacian operators are all defined using central differences,

$$\boldsymbol{G}\varphi = \frac{\varphi_{i+1,j} - \varphi_{i,j}}{\Delta x}\boldsymbol{e}_x + \frac{\varphi_{i,j+1} - \varphi_{i,j}}{\Delta y}\boldsymbol{e}_y, \tag{3.16}$$

$$\boldsymbol{D}\boldsymbol{w} = \frac{w^1_{i+1/2,j} - w^1_{i-1/2,j}}{\Delta x} + \frac{w^2_{i,j+1/2} - w^2_{i,j-1/2}}{\Delta y}, \tag{3.17}$$

$$\begin{aligned}\boldsymbol{L}\boldsymbol{w} = &\left(\frac{w^1_{i+3/2,j} - 2w^1_{i+1/2,j} + w^1_{i-1/2,j}}{(\Delta x)^2} + \frac{w^1_{i,j+3/2} - 2w^1_{i,j+1/2} + w^1_{i,j-1/2}}{(\Delta y)^2}\right)\boldsymbol{e}_x \\ &+ \left(\frac{w^2_{i+3/2,j} - 2w^2_{i+1/2,j} + w^2_{i-1/2,j}}{(\Delta x)^2} + \frac{w^2_{i,j+3/2} - 2w^2_{i,j+1/2} + w^2_{i,j-1/2}}{(\Delta y)^2}\right)\boldsymbol{e}_y.\end{aligned} \tag{3.18}$$

This is consistent with the staggered grid, since the gradient and Laplacian returns a vector with components defined on the cell faces, while the divergence returns a scalar defined at the cell centres. These operators are all second-order approximations to their continuous counterparts.

3.3.2 The convective operator

Both the Navier-Stokes equations and the level-set equation involves the convective derivative,

$$\frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}, \tag{3.19}$$

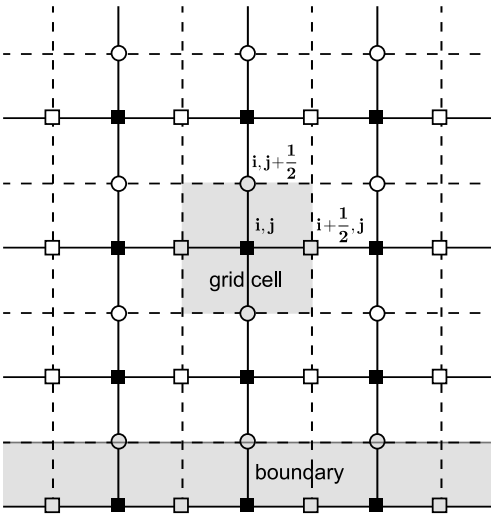


Figure 3.1: Example of a staggered grid. The scalar values are stored at the cell centres (i, j), while the x-component and y-component of vectors are stored at the cell faces, $(i+1/2, j)$ and $(i, j+1/2)$, respectively.

where the convective operator is $\boldsymbol{u}\cdot\boldsymbol{\nabla}$. Consider the level-set equation in the one-dimensional case,

$$\frac{\partial\phi}{\partial t}+u\phi_x=0, \tag{3.20}$$

where u is the velocity and $u\phi_x$ is the convective term, which represents the direction and speed of the interfacial advection. Central differences are not recommended for the discretization of this term, since they tend to introduce spurious oscillations. Instead, the spatial discretization should depend on the direction of the velocity. If $u<0$, the spatial discretization should use information to the right of the current node, and if $u>0$, the information to the left should be used.

In this thesis, the WENO scheme is used for the discretization of the convective operator. This scheme obtains fifth-order accuracy in smooth regions and third-order accuracy in nonsmooth regions. Following is a short derivation of the one dimensional WENO scheme. The one-dimensional approach can be used independently for each dimension when expanding to two and three dimensions. For further details on the subject, see Osher & Fedkiw (2003).

As already argued, the sign of u determines the spatial discretization of φ_x, i.e.

$$\frac{\partial\varphi}{\partial x}\simeq\begin{cases}\varphi_x^+ & \text{if } u\leq 0\\ \varphi_x^- & \text{if } u>0\end{cases}. \tag{3.21}$$

Let $D^+\varphi$ and $D^-\varphi$ denote the standard forward difference and backward difference respectively,

$$D^+\varphi_i=\frac{\varphi_{i+1}-\varphi_i}{\Delta x}, \tag{3.22}$$

$$D^-\varphi_i=\frac{\varphi_i-\varphi_{i-1}}{\Delta x}, \tag{3.23}$$

where $\varphi_i=\varphi(x_i)$ denotes the arbitrary function evaluated at x_i. Letting $\varphi_x^+=D^+\varphi$ and $\varphi_x^-=D^-\varphi$ gives the method called upwind differencing. The upwind scheme is only a first-order method, and can be improved upon by using more accurate approximations for φ_x^+ and φ_x^-. The essential non-oscillatory (ENO) correction scheme is a third-order method which utilizes a subset of

$$\{\varphi_{k-3},\varphi_{k-2},\varphi_{k-1},\varphi_k,\varphi_{k+1},\varphi_{k+2}\},$$

where $k=i$ for φ_x^- and $k=i+1$ for φ_x^+. Since the WENO scheme is an improvement of the ENO scheme, it is natural to start with a derivation of the ENO scheme.

The four first divided differences of φ are

$$D_i^0\varphi=\varphi_i, \tag{3.24}$$

$$D_{i+1/2}^1\varphi=\frac{D_{i+1}^0\varphi-D_i^0\varphi}{\Delta x}, \tag{3.25}$$

$$D_i^2\varphi=\frac{D_{i+1/2}^1\varphi-D_{i-1/2}^1\varphi}{2\Delta x}, \tag{3.26}$$

$$D_{i+1/2}^3\varphi=\frac{D_{i+1}^2\varphi-D_i^2\varphi}{3\Delta x}. \tag{3.27}$$

Note that the zeroth and the second divided differences are defined at the grid nodes, while the first and the third are defined at the midpoints between grid nodes. The divided differences will be used to construct a polynomial of the form

$$\varphi(x)=\sum_{j=0}^{3}Q_j(x), \tag{3.28}$$

where Q_j is a polynomial of degree j. Differentiating and evaluating at x_i yields

$$\varphi_x(x_i) = \sum_{j=1}^{3} Q_j'(x). \tag{3.29}$$

By using appropriate representations of the Q_j's, equation (3.29) can be used to approximate φ_x^+ or φ_x^-. Beginning with the first-order term, let

$$Q_1(x) = (D^1_{k+1/2}\varphi)(x - x_i), \tag{3.30}$$

such that

$$Q_1'(x_i) = D^1_{k+1/2}\varphi. \tag{3.31}$$

This gives an approximation to φ_x^- or φ_x^+, depending on whether $k = i-1$ or $k = i$. Note that the first-order polynomial is exactly the first-order upwind differencing. To improve the approximation, let $\xi_1 = |D^2_{k+1}\varphi| - |D^2_k\varphi|$, and define

$$C = \begin{cases} D^2_k\varphi & \text{if } \xi_1 \geq 0 \\ D^2_{k+1}\varphi & \text{otherwise} \end{cases}, \tag{3.32}$$

$$k^* = \begin{cases} k-1 & \text{if } \xi_1 \geq 0 \\ k & \text{otherwise} \end{cases}. \tag{3.33}$$

Now the second-order correction can be found by

$$Q_2(x) = C(x - x_k)(x - x_{k+1}), \tag{3.34}$$

and

$$Q_2'(x_i) = C(2(i-k) - 1)\Delta x. \tag{3.35}$$

The third-order correction can be found in a similar way; let $\xi_2 = |D^3_{k^*+3/2}\varphi| - |D^3_{k^*+1/2}\varphi|$, and let

$$C^* = \begin{cases} D^3_{k^*+3/2}\varphi & \text{if } \xi_2 \geq 0 \\ D^3_{k^*+1/2}\varphi & \text{otherwise} \end{cases}. \tag{3.36}$$

The third polynomial becomes

$$Q_3(x) = C^*(x - x_{k^*})(x - x_{k^*+1})(x - x_{k^*+2}), \tag{3.37}$$

and

$$Q_3'(x_i) = C^*(3(i-k^*)^2 - 6(i-k^*) + 2)(\Delta x)^2. \tag{3.38}$$

These corrections can be used to find third-order approximations for φ_x^+ and φ_x^- by inserting equations (3.31), (3.35) and (3.38) into equation (3.29). There are three possible approximations for each of $(\varphi_x^+)_i$ and $(\varphi_x^-)_i$. Defining $v_j = D^-\varphi_{i-3+j}$ for $j = 1,2,3,4$ and 5 gives the three approximations for $(\varphi_x^-)_i$:

$$\varphi_x^1 = \frac{v_1}{3} - \frac{7v_2}{6} + \frac{11v_3}{6}, \tag{3.39}$$

$$\varphi_x^2 = -\frac{v_2}{6} + \frac{5v_3}{6} + \frac{v_4}{3}, \tag{3.40}$$

$$\varphi_x^3 = \frac{v_3}{3} - \frac{5v_4}{6} - \frac{v_5}{6}. \tag{3.41}$$

The same three formulae gives $(\varphi_x^+)_i$ by defining $v_j = D^+\varphi_{i+3-j}$.

The ENO-scheme chooses one of three approximations for each of $(\varphi_x^+)_i$ and $(\varphi_x^-)_i$, depending on the value of the divided difference. In the WENO-scheme, a weighted combination of these three is used, giving fifth-order accuracy in smooth regions. The weighted combination can be written as

$$\varphi_x = \omega_1 \varphi_x^1 + \omega_2 \varphi_x^2 + \omega_3 \varphi_x^3, \tag{3.42}$$

where $0 \leq \omega_k \leq 1$ are weights such that $\sum_{k=1}^{3} \omega_k = 1$. In Jiang & Shu (1996) it was shown that the optimal fifth-order accuracy can be achieved using $\omega_1 = 0.1, \omega_2 = 0.6$ and $\omega_3 = 0.3$. In nonsmooth regions, however, this weighting can be very inaccurate. In Jiang & Peng (2000) a smoothness estimation was used to improve the order in nonsmooth regions. The smoothness of the stencils given in equations (3.39) to (3.41) are found as

$$S_1 = \frac{13}{12}(v_1 - 2v_2 + v_3)^2 + \frac{1}{4}(v_1 - 4v_2 + 3v_3)^2, \tag{3.43}$$

$$S_2 = \frac{13}{12}(v_2 - 2v_3 + v_4)^2 + \frac{1}{4}(v_2 - v_4)^2, \tag{3.44}$$

$$S_3 = \frac{13}{12}(v_3 - 2v_4 + v_5)^2 + \frac{1}{4}(3v_3 - 4v_4 + v_5)^2, \tag{3.45}$$

respectively. Now define

$$\alpha_1 = \frac{0.1}{(S_1 + \varepsilon)^2}, \tag{3.46}$$

$$\alpha_2 = \frac{0.6}{(S_2 + \varepsilon)^2} \tag{3.47}$$

and

$$\alpha_3 = \frac{0.1}{(S_3 + \varepsilon)^2}, \tag{3.48}$$

where $\varepsilon = 10^{-6}$ is used to prevent division by zero. The αs are normalized, giving the following weights:

$$\omega_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3}, \tag{3.49}$$

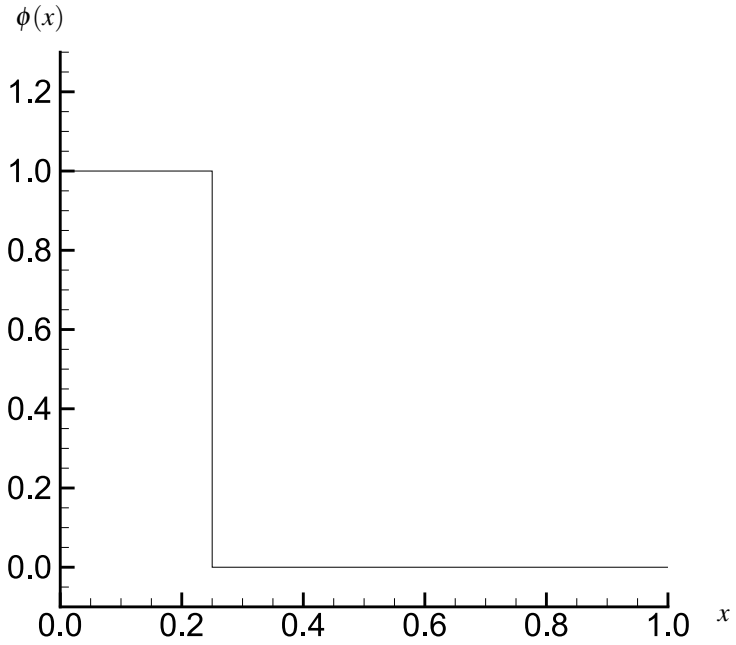
$$\omega_2 = \frac{\alpha_2}{\alpha_1 + \alpha_2 + \alpha_3}, \tag{3.50}$$

$$\omega_3 = \frac{\alpha_3}{\alpha_1 + \alpha_2 + \alpha_3}. \tag{3.51}$$

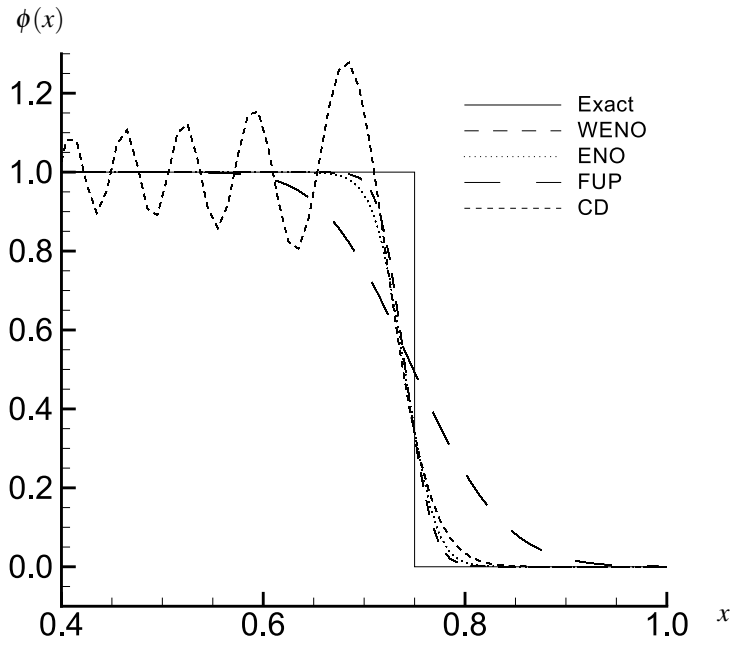
In a smooth region, the weights will be very close to the optimal weights and result in fifth-order accuracy. In nonsmooth regions, the stencils that represent the least smooth areas gets small values of α, reducing their influence on the approximation. This way the WENO scheme achieves third-order accuracy in nonsmooth regions. Also, note that the number of calculations needed for solving the WENO scheme is small compared to that of the ENO scheme, so the WENO scheme is in general much faster than the ENO scheme.

A simple test case demonstrates the difference in how the schemes perform. Let $\phi(t,x)$ be initialized to the step function shown in Figure 3.2(a), and let the advection speed u be constant, $u = 0.5$. Figure 3.2(b) shows the step function after $t = 1$ s. The solid line shows the exact solution, the medium-dashed line shows the WENO solution, the dotted line shows the ENO solution, the short-dashed line shows the central-difference (CD) solution and the long-dashed line shows the first-order upwind (FUP) solution. All the simulations were solved on a grid with $\Delta x = 0.01$ using a four-stage third order SSP RK scheme with CFL number $C = 0.2$[1]. It can be seen that CD

[1] See Section 3.5 for a review of the SSP RK schemes and an explanation of the CFL number



(a) Initial step function, $\phi(x)$.



(b) Step function at $t = 1\,\mathrm{s}$.

Figure 3.2: A comparison of the different discretization methods for the convective term. (a) shows the initial step function and (b) shows the result at $t = 1\,\mathrm{s}$. The solid line is the exact solution, the medium-dashed line is the WENO solution, the dotted line is the ENO solution, the short-dashed line is the central-difference (CD) solution and the long-dashed line is the first-order upwind (FUP) solution.

Table 3.1: Simulation run times for the different discretization schemes.

Discretization scheme	Run time (s)
WENO	13.42
ENO	43.02
CD	8.99
FUP	8.64

introduces spurious oscillations, and that FUP smears the solution. The smearing is also present in ENO and WENO, but they perform noticeably better than FUP. The difference between WENO and ENO is small, which is expected since they are both third-order methods in nonsmooth areas. The test also demonstrated that the calculation time using WENO was considerably less than the corresponding time for ENO, see Table 3.1.

3.4 The ghost-fluid method

In Section 2.2, the interfacial discontinuities due to the interfacial force and the fluid properties were discussed. The ghost-fluid method (GFM) is used to treat these interfacial discontinuities numerically by modifying the discretization schemes close to the interface. This modification results in a first-order scheme at the interface. The method was first introduced by Fedkiw (1999), and was originally developed for treating discontinuities in compressible flow. In Kang *et al.* (2000), it was modified for incompressible flow.

Equation (2.39) shows that the viscous term, $\mu\boldsymbol{\nabla u}$, is discontinuous across the interface. To solve the flow equations, terms like

$$\boldsymbol{\nabla}\cdot\mu\boldsymbol{\nabla u} \tag{3.52}$$

must be addressed. The simplification

$$\boldsymbol{\nabla}\cdot\mu\boldsymbol{\nabla u} = \mu\boldsymbol{\Delta u} \tag{3.53}$$

follows when μ is constant, but this simplification leads to numerical smearing across the interface where $\mu\boldsymbol{\nabla} u$ is discontinuous. Following is an example of how the GFM is used to discretize the term (3.52) in the one-dimensional case,

$$\frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial x}\right). \tag{3.54}$$

First, let the tensor $\boldsymbol{J}$ be a discretization of the expressions for the jump conditions such that

$$\boldsymbol{J} \simeq [\mu\boldsymbol{\nabla} u], \tag{3.55}$$

i.e. $J^{11} \simeq [\mu\,\partial u/\partial x]$. Now consider the discretization of the viscous term (3.54) at the point $x_{i+1/2,j}$. Define

$$u_L = u_{i-1/2,j}, \tag{3.56}$$

$$u_M = u_{i+1/2,j}, \tag{3.57}$$

$$u_R = u_{i+3/2,j}. \tag{3.58}$$

Also, let the averaged values of J^{11} and the level-set function ϕ at the corresponding points be ϕ_L, ϕ_M, ϕ_R, J_L, J_M and J_R. The simplest cases are when the interface is not close enough to be

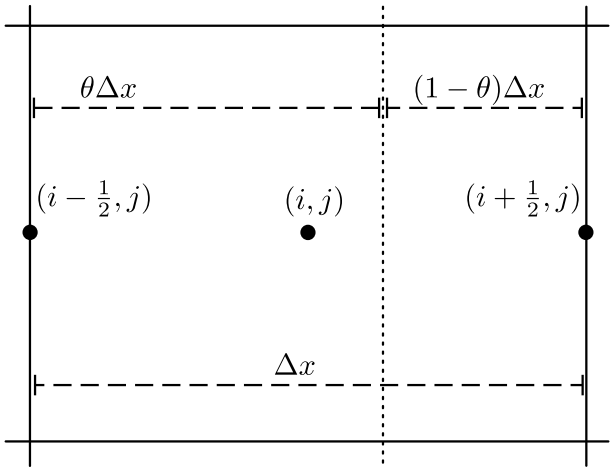


Figure 3.3: The split cell (i, j), where the dotted vertical line depicts the interface.

considered, that is ϕ_L, ϕ_M and ϕ_R are all either greater than or equal to zero or less than zero. In these cases,

$$\left(\frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial x}\right)\right)_{i+1/2,j} \simeq \mu^{+}\frac{u_R - 2u_M + u_L}{(\Delta x)^2}, \tag{3.59}$$

or

$$\left(\frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial x}\right)\right)_{i+1/2,j} \simeq= \mu^{-}\frac{u_R - 2u_M + u_L}{(\Delta x)^2}, \tag{3.60}$$

respectively. Otherwise, let

$$\left(\frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial x}\right)\right)_{i+1/2,j} \simeq \frac{\left(\mu\,\frac{\partial u}{\partial x}\right)_R - \left(\mu\,\frac{\partial u}{\partial x}\right)_L}{\Delta x}. \tag{3.61}$$

Suppose that $\phi_L < 0$ and $\phi_M \geq 0$, such that the interface splits the cell (i, j) into two pieces of size $\theta\Delta x$ on the left side and $(1-\theta)\Delta x$ on the right side, where

$$\theta = \frac{|\phi_L|}{|\phi_L| + |\phi_M|}, \tag{3.62}$$

see Figure 3.3. Let u_I denote the continuous velocity at the interface, and calculate the jump at the interface,

$$J_I = \theta J_M + (1-\theta)J_L. \tag{3.63}$$

Now discretize the jump condition $[\mu\,\partial u/\partial x]$ at the interface to obtain

$$\mu^{+}\left(\frac{u_M - u_I}{(1-\theta)\Delta x}\right) - \mu^{-}\left(\frac{u_I - u_L}{\theta\Delta x}\right) = J_I, \tag{3.64}$$

and solve this equation for u_I,

$$u_I = \frac{\mu^{+}u_M\theta + \mu^{-}u_L(1-\theta) - J_I\theta(1-\theta)\Delta x}{\mu^{+}\theta + \mu^{-}(1-\theta)}. \tag{3.65}$$

The expression for u_I can be used to discretize the split cell, i.e.

$$\left(\mu\frac{\partial u}{\partial x}\right)_L \simeq \mu^{+}\left(\frac{u_M - u_I}{(1-\theta)\Delta x}\right) = \hat{\mu}\left(\frac{u_M - u_L}{\Delta x}\right) + \frac{\hat{\mu}J_I\theta}{\mu^{-}}, \tag{3.66}$$

where $\hat{\mu}$ defines an effective μ,

$$\hat{\mu} = \frac{\mu^+\mu^-}{\mu^+\theta + \mu^-(1-\theta)}. \tag{3.67}$$

Similarly one finds when $\phi_L \geq 0$ and $\phi_M < 0$,

$$\left(\mu\frac{\partial u}{\partial x}\right)_L \simeq \mu^- \left(\frac{u_M - u_I}{(1-\theta)\Delta x}\right) = \tilde{\mu}\left(\frac{u_M - u_L}{\Delta x}\right) + \frac{\tilde{\mu}J_I\theta}{\mu^+}, \tag{3.68}$$

where

$$\tilde{\mu} = \frac{\mu^+\mu^-}{\mu^-\theta + \mu^+(1-\theta)}. \tag{3.69}$$

For the two remaining cases, let

$$\theta = \frac{|\phi_R|}{|\phi_R| + |\phi_M|} \tag{3.70}$$

and

$$J_I = \theta J_M + (1-\theta)J_R. \tag{3.71}$$

The same approach can be used, and one finds that when $\phi_R \geq 0$ and $\phi_M < 0$,

$$\left(\mu\frac{\partial u}{\partial x}\right)_R \simeq \mu^- \left(\frac{u_I - u_M}{(1-\theta)\Delta x}\right) = \tilde{\mu}\left(\frac{u_R - u_M}{\Delta x}\right) + \frac{\tilde{\mu}J_I\theta}{\mu^+}, \tag{3.72}$$

and when $\phi_R < 0$ and $\phi_M \geq 0$,

$$\left(\mu\frac{\partial u}{\partial x}\right)_R \simeq \mu^+ \left(\frac{u_I - u_M}{(1-\theta)\Delta x}\right) = \hat{\mu}\left(\frac{u_R - u_M}{\Delta x}\right) + \frac{\hat{\mu}J_I\theta}{\mu^-}. \tag{3.73}$$

For the interested reader, further details regarding the ghost-fluid method used on the Poisson equation can be found in Hansen (2005) and Osher & Fedkiw (2003) or in the articles Fedkiw (1999) and Kang *et al.* (2000).

3.5 Temporal discretization

All the partial differential equations used in this thesis are semi-discretized, resulting in systems of ODEs that may be written as

$$\frac{\mathrm{d}\varphi(t)}{\mathrm{d}t} = f(\varphi(t)) \quad (t \geq 0), \qquad \varphi(0) = \varphi_0, \tag{3.74}$$

where $\varphi(t)$ is once again used as an arbitrary, smooth function and φ_0 is given. The ODEs are discretized using one of the strong stability preserving Runge-Kutta (SSP RK) schemes. The SSP RK schemes are used because they do not introduce instabilities in the solution. This is done by extending low-order methods to higher order while preserving the stability of the low-order method, see Ketcheson & Robinson (2005) or Hansen (2005). The SSP RK schemes used in this thesis also allows for some memory optimizations. Details on the general RK method can be found in most books on numerical analysis, i.e. Iserles (1996) or Burden & Faires (2004).

This section is divided into two parts. The relevant schemes will be presented in the first part, and the time-step restriction will be discussed in the second part.

$$\begin{array}{c|ccc} 0 & & & \\ c_2 & a_{21} & & \\ \vdots & \vdots & \ddots & \\ c_s & a_{s1} & \cdots & a_{s(s-1)} \\ \hline & b_1 & \cdots & b_{s-1} \quad b_s \end{array}$$

(a) The general explicit Butcher table.

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array}$$

(b) Euler method

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ 1/2 & 1/4 & 1/4 \\ \hline & 1/6 & 1/6 \quad 2/3 \end{array}$$

(c) Three-stage third-order SSP RK

$$\begin{array}{c|ccc} 0 & & & \\ 1/3 & 1/3 & & \\ 2/3 & 1/3 & 1/3 & \\ 1 & 1/3 & 1/3 & 1/3 \\ \hline & 1/4 & 1/4 & 1/4 \quad 1/4 \end{array}$$

(d) Four-stage second-order SSP RK

$$\begin{array}{c|ccc} 0 & & & \\ 1/2 & 1/2 & & \\ 1 & 1/2 & 1/2 & \\ 1/2 & 1/6 & 1/6 & 1/6 \\ \hline & 1/6 & 1/6 & 1/6 \quad 1/2 \end{array}$$

(e) Four-stage third-order SSP RK

Figure 3.4: Butcher tables for the relevant SSP RK schemes.

3.5.1 The SSP RK schemes

Any explicit s-stage RK scheme for the problem (3.74) can be written as

$$\varphi^{(i)} = \varphi_n + \Delta t \sum_{j=1}^{s} a_{ij} f\left(\varphi^{(j)}\right) \quad (i = 1,2,\ldots,s), \tag{3.75}$$

$$\varphi_{n+1} = \varphi_n + \Delta t \sum_{i=1}^{s} b_i f\left(\varphi^{(i)}\right), \tag{3.76}$$

where $a_{ij} = 0$ for $j \geq i$. The different schemes can be represented by the corresponding Butcher tables, see Figure 3.4(a). The Euler method is shown as an example in Figure 3.4(b). In the numerical framework, several SSP RK schemes are implemented. The most frequently used schemes are a three-stage third-order scheme, a four-stage third-order scheme and a four-stage second-order scheme, represented with Butcher tables in Figures 3.4(c), 3.4(e) and 3.4(d), respectively. If not otherwise mentioned, the four-stage third-order scheme is used for the equations of motion and the four-stage second-order scheme is used for the level-set equations. The higher-stage schemes are stable for larger time steps than lower-stage schemes, at least on linear problems.

It should be noted that the implementation of the RK schemes used in this work are based on the Shu-Osher form of the RK method, see Shu & Osher (1988),

$$\varphi^{(0)} = \varphi_n \tag{3.77}$$

$$\varphi^{(i)} = \sum_{j=0}^{i-1} \left(\alpha_{ij}\varphi^{(j)} + \Delta t \beta_{ij} f(\varphi^{(j)})\right), \qquad i = 1,\ldots,s \tag{3.78}$$

$$\varphi_{n+1} = \varphi^{(s)}. \tag{3.79}$$

It can be shown that the SSP property is enforced when $\alpha_{ij} \geq 0$ and $\beta_{ij} \geq 0$, and the schemes used in this work are all optimal SSP schemes, see Ketcheson & Robinson (2005). The four-stage

second-order scheme is given in its Shu-Osher form as an example:

$$\varphi^{(1)} = \varphi_n + \frac{\Delta t}{3} f(\varphi_n) \tag{3.80}$$

$$\varphi^{(2)} = \varphi^{(1)} + \frac{\Delta t}{3} f(\varphi^{(1)}) \tag{3.81}$$

$$\varphi^{(3)} = \varphi^{(2)} + \frac{\Delta t}{3} f(\varphi^{(2)}) \tag{3.82}$$

$$\varphi_{n+1} = \frac{1}{4}\varphi_n + \frac{3}{4}\varphi^{(3)} + \frac{\Delta t}{4} f(\varphi^{(3)}). \tag{3.83}$$

For more details on the Shu-Osher form, the SSP property and optimal SSP schemes, the reader is reffered to Ferracina & Spijker (2004) and Ferracina & Spijker (2005).

3.5.2 The time-step restriction

Stability of the explicit time-integration method can be enforced using the Courant-Friedrich-Lewy (CFL) condition. Simply put, this condition asserts that a fluid particle does not move more than one grid cell during one time step. In one dimension, this can be written as

$$|u|\Delta t < \Delta x \tag{3.84}$$

where Δt is the time-step length, Δx is the grid size and u is the fluid velocity. This leads to the CFL time-step restriction

$$\Delta t \leq \frac{\Delta x}{\max\{|u|\}}, \tag{3.85}$$

where $\max\{|u|\}$ is the largest value of $|u|$ on the entire grid. The equation is usually enforced by using a CFL number C such that

$$\Delta t \left(\frac{\Delta x}{\max\{|u|\}} \right) = C, \tag{3.86}$$

where $0 < C < 1$. In two dimensions, the CFL condition can be written as

$$\Delta t \max \left\{ \frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} \right\} = C, \tag{3.87}$$

where u and v represents the different components of the flow velocity. The CFL condition must be generalized to take into account the effects of viscous stress, surface tension and gravity. In Kang *et al.* (2000), a method to compute the criterion including these effects is given. In the implemented framework, a variant of this is used. Let C_c, C_v, C_s and C_g be the contribution from from the convective term, the viscous stresses, the surface tension and the gravity, respectively. Then the time-step restriction is defined as

$$\Delta t = \frac{C}{(C_c + C_v)/2 + \sqrt{(C_c + C_v)^2 + 4C_g^2 + 4C_s^2}}, \tag{3.88}$$

where

$$C_c = \frac{\max\{|u|\}}{\Delta x} + \frac{\max\{|v|\}}{\Delta y}, \tag{3.89}$$

$$C_v = 2\max\left\{\nu^-, \nu^+\right\}\left(\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2}\right), \tag{3.90}$$

$$C_s = \sqrt{\frac{\sigma_{\max}|\kappa_{\max}|}{\max\{\rho^+, \rho^-\}(\min\{\Delta x, \Delta y\})^2}}, \tag{3.91}$$

$$C_g = \sqrt{\frac{|g_x|}{\Delta y} + \frac{|g_y|}{\Delta y}}. \tag{3.92}$$

In Kang *et al.* (2000), the minimum and not the maximum density was used in equation (3.91). Because of this, the current criterion is less conservative in flows where there is a large density ratio between the phases. Also note that for cases where the surface tension varies along the interface, the maximum of the surface tension over the interface is used, $\sigma_{\max}$.

3.6 Implementation details

In this section, some important implementation details will be discussed. The discretization and implementation of the Marangoni terms will be described first, followed by a presentation of the boundary conditions used in this thesis.

3.6.1 The Marangoni term

In the old implementation, the surface tension was assumed to be constant. The first step of removing this restriction is to let the surface tension be a varying variable on the interface. This is handled by letting the coefficient of surface tension be represented by a field variable over the entire domain. Since the surface tension is only defined on the interface, the value of the surface tension at the interface is found by interpolation.

The next step is to add the Marangoni terms to the routine that calculates the jump conditions for the viscous stress tensor. In the implemented framework, the tensor $\boldsymbol{J}$ from equation (3.55) is a straightforward discretization of equation (2.35), where the last term $(\boldsymbol{t}\cdot\boldsymbol{f}_{\text{sfd}})\boldsymbol{tn}$ is ignored. When the surface tension varies, this term becomes $(\boldsymbol{t}\cdot\boldsymbol{\nabla}_s\sigma)\boldsymbol{tn}$ and results in equations (2.43) to (2.46). The surface tension terms are discretized with central differences, e.g.

$$\left.\frac{\partial\sigma}{\partial x}\right|_{i+1/2} \simeq \frac{\sigma_{i+1} - \sigma_i}{\Delta x}. \tag{3.93}$$

3.6.2 Boundary conditions

Several types of boundary conditions are used in this work. Following is a short description of each of them, based on Griebel *et al.* (1998).

First, let computational grid have $i_{\max} \times j_{\max}$ internal nodes and a set of up to three ghost cells across the boundary in each direction. Let n denote the number of ghost cells in each direction, and

let $k = 1, \ldots, n$. Also, note that for the no-slip and the free-slip condition, the level-set function is reflected across the boundary using

$$\begin{aligned} \phi_{i,1-k} &= \phi_{i,k}, \quad \phi_{i,j_{\max}+k} = \phi_{i,j_{\max}+1-k}, \quad i = 1, \ldots, i_{\max} - 1, \\ \phi_{1-k,j} &= \phi_{k,j}, \quad \phi_{i_{\max}+k,j} = \phi_{i_{\max}+1-k,j}, \quad j = 1, \ldots, j_{\max} - 1. \end{aligned} \tag{3.94}$$

No-slip The no-slip condition on the boundary implies that the fluid velocity relative to the boundary is zero at the boundary. This means that both the normal and the tangent components of the fluid velocity at the boundary is zero. The normal components of the velocity are given at the walls,

$$\begin{aligned} u_{0,j} = u_{i_{\max},j} &= 0, \qquad j = 1, \ldots, j_{\max}, \\ v_{i,0} = v_{i,j_{\max}} &= 0, \qquad i = 1, \ldots, i_{\max}. \end{aligned} \tag{3.95}$$

The tangential components of the velocities do not lie on the boundary, so the no-slip condition can be enforced with

$$\begin{aligned} u_{i,1-k} &= -u_{i,k}, \quad u_{i,j_{\max}+k} = -u_{i,j_{\max}+1-k}, \quad i = 1, \ldots, i_{\max} - 1, \\ v_{1-k,j} &= -v_{k,j}, \quad v_{i_{\max}+k,j} = -v_{i_{\max}+1-k,j}, \quad j = 1, \ldots, j_{\max} - 1, \end{aligned} \tag{3.96}$$

such that the average of the values on either side of the boundary is zero.

Free-slip With free-slip conditions on the boundary, no fluid can pass through the boundary, but there is no friction. This means that the velocity components normal to the boundary is zero, see (3.95). Also, the normal derivative of the velocity tangent to the boundary is zero, which is achieved by mirroring the velocities of the computational domain across the border:

$$\begin{aligned} u_{i,1-k} &= u_{i,k}, \quad u_{i,j_{\max}+k} = u_{i,j_{\max}+1-k}, \quad i = 1, \ldots, i_{\max} - 1, \\ v_{1-k,j} &= v_{k,j}, \quad v_{i_{\max}+k,j} = v_{i_{\max}+1-k,j}, \quad j = 1, \ldots, j_{\max} - 1. \end{aligned} \tag{3.97}$$

Periodic When using periodic boundaries, the ends of the domain are overlapping. Thus periodic boundaries always come in pairs, i.e. if the west boundary is periodic, then obviously the east boundary is periodic as well. Also, the overlapping of the domain results in a smaller computational domain, so the width of the domain must be chosen to exceed the period length by one grid-cell length. The periodic boundary condition will be given for vertical boundaries only, as the expressions for horizontal boundaries are equivalent. The periodic condition for the horizontal velocity-component is

$$\begin{aligned} u_{1-k,j} &= u_{i_{\max}-k,j}, \\ u_{i_{\max}-1+k,j} &= u_{k,j}, \end{aligned} \tag{3.98}$$

where $j = 1, \ldots, j_{\max}$. The expression for the vertical velocity-component is

$$\begin{aligned} v_{1,j} &= v_{i_{\max},j}, \\ v_{1-k,j} &= v_{i_{\max}-k,j}, \\ v_{i_{\max}+k,j} &= v_{1+k,j}, \end{aligned} \tag{3.99}$$

where $i = 1, \ldots, i_{\max}$. Since the level-set function is defined at the cell center, the expression for the level-set function is the same as the expression for the vertical velocity-components.

It should be noted that for a periodic domain, the pressure should also be periodic. Since a periodic boundary condition for the pressure is not implemented, the pressure is restricted to being constant in the direction of periodicity.

3.7 Summary

In this chapter, the discretization and numerical methods used to solve the mathematical model given in Chapter 2 have been discussed. First, the projection method was discussed. It was shown how the semi-discretized governing equations could be solved as a system of ODEs coupled with a Poisson equation. Then the level-set method was reviewed, and it was shown how it can be used to implicitly capture the interface in two-phase simulations. Next, the spatial operators were discussed, and the approximation of the convective term was discussed in detail. The third-order ENO-scheme and the fifth-order WENO-scheme were reviewed, and an example was made to demonstrate the performence of the different schemes for the advection of a step function. The example showed that the WENO scheme is preferable for the approximation of the convective operator, both with regard to accuracy, and with regard to computational time. Then the treatment of the interfacial discontinuities using the ghost-fluid method was discussed, followed by a section dedicated to the temporal discretization. Finally, some details concerning the implementation of the boundary conditions and the Marangoni effect were given.

4 Analysis of the Marangoni force

The mathematical model has been derived and implemented under the assumption that the surface tension may vary along the interface. It has been shown that this leads to Marangoni terms in the jump condition for the viscous stress, see Section 2.2 and equations (2.43) to (2.46). The implementation of these terms were discussed in Section 3.6.

The Marangoni terms induce the so-called Marangoni effect, which will be discussed in this chapter. Two problems will be analysed and tested numerically, and will be used to verify that the implementation of the Marangoni terms is reliable.

4.1 The Marangoni effect

The derivation of the jump conditions in Section 2.2 showed that gradients in the surface tension induce forces that modify the interface boundary conditions. The effect of these forces is called the Marangoni effect. It is easy to show that a two-phase flow where there is a non-zero Marangoni force can not be static. The flow is static when $\boldsymbol{u} = 0$, which means that $\boldsymbol{\nabla u} = 0$ and

$$[\boldsymbol{\nabla u}] = 0, \tag{4.1}$$

but if $\boldsymbol{\nabla}_s \sigma \neq 0$, then equation (2.39) can be used to show that

$$[\boldsymbol{\nabla u}] = (\boldsymbol{t} \cdot \boldsymbol{\nabla}_s \sigma)\, \boldsymbol{tn} \neq 0. \tag{4.2}$$

Several different factors may lead to gradients in the surface tension, such as gradients in the temperature, surfactant concentration and electric potential, see Brennen (2005, Section 3.3). Here it is also stated that for most ranges of temperature, the surface tension decreases linearly with temperature, reaching zero at the critical point.

The effects of a non-uniform surfactant distribution will be discussed in detail in Chapter 5. Since this chapter will describe the Marangoni effect, it can be viewed as a preparation for the study of surfactant-influenced flows.

In the following sections, the coefficient of surface tension will be given as a function of space, e.g.

$$\sigma(x) = \sigma_0 + \sigma_1 x, \tag{4.3}$$

where σ_0 and σ_1 are constants and the spatial parameter x is to be evaluated on the interface. A variation in temperature could be used as an explanation of the variation in surface tension, but then it should be noted that all other temperature effects are still neglected and that the temperature is assumed to remain constant in time. Thus the results from the following test cases will not be physically plausible, but they should show the correct mathematical behaviour and illustrate the Marangoni effect to a two-phase flow.

4.2 Plane interfaces

4.2.1 A horizontal interface

Consider a two-phase flow in an infinite-length tube of height $2h$, with an interface at height h. Gravity effects are neglected, and the densities and viscosities are assumed equal, i.e. $\rho_1 = \rho_2 = \rho$ and $\mu_1 = \mu_2 = \mu$. Let the coefficient of surface tension increase linearly along the interface,

$$\sigma(x) = \sigma_0 + \sigma_1 x, \tag{4.4}$$

such that

$$\frac{\partial \sigma}{\partial x} = \sigma_1. \tag{4.5}$$

The set-up is depicted in Figure 4.1. This problem is actually inspired by the lid-driven cavity problem, see e.g. Ghia *et al.* (1982). A lid moving with constant velocity instead of the interface would have the same effect as the linear surface-tension gradient.

Consider the described flow when it has reached steady state, i.e. when

$$\frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = 0, \tag{4.6}$$

$$\frac{\partial u}{\partial x} = 0, \tag{4.7}$$

$$v = 0, \tag{4.8}$$

$$\boldsymbol{\nabla} p = 0, \tag{4.9}$$

where $\boldsymbol{u} = (u, v)$ is the velocity vector and u and v are the horizontal and vertical velocities, respectively. The velocity at the top and bottom boundaries are zero, that is

$$u(0) = u(2h) = 0. \tag{4.10}$$

The jump in the viscous stress tensor is given by equation (2.39), and the only non-zero term is

$$\left[\mu \frac{\partial u}{\partial y}\right] = \mu \left(\frac{\partial u_2}{\partial y} - \frac{\partial u_1}{\partial y} \right) = -\sigma_1. \tag{4.11}$$

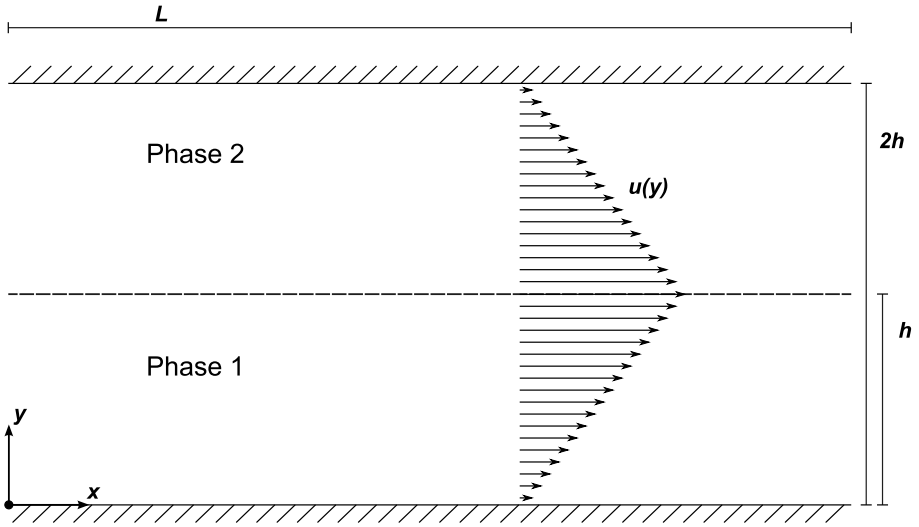


Figure 4.1: A sketch of the simulation set-up used to test the Marangoni effect with a horizontal interface.

The given assumptions and boundary conditions imply that the steady-state solution of the horizontal velocity is

$$u(y) = u_{\max}\left(1 - \frac{|y-h|}{h}\right), \tag{4.12a}$$

where

$$u_{\max} = u(h) = \frac{h\sigma_1}{2\mu}. \tag{4.12b}$$

Equations (4.12) show that a variation in the surface tension along an interface results in a flow in the direction of increasing surface tension.

In the simulations, the set-up shown in Figure 4.1 is used with $L = 2h$. All simulations are calculated on a 129×129 grid and run until $t = 20\,\text{s}$. The parameters used are

$$\mu = 0.002\ \text{Pa s}, \qquad \sigma_1 = 0.01\ \text{N/m}^2, \qquad h = 0.4\ \text{m},$$

which correspond to the following analytic solution for the flow, see equations (4.12):

$$u(y) = 1 - \frac{|y-0.4|}{0.4}, \tag{4.13}$$

where the velocity units are omitted. To simulate an infinite length, periodic boundaries are used for the left and right borders. The implementation of the periodic boundaries is only valid for cases with constant pressure, see the paragraph about the implementation of periodic boundaries in Section 3.6. If the initial state is the analytic solution, the pressure will remain constant, and the stationary state should be kept in time. If the initial state is not the stationary state, however, the pressure will not be constant and errors due to the boundary conditions will ruin the simulation.

Figure 4.2 shows the flow-velocity profiles for two simulations. The solid lines show the initial profile and the dotted lines show the profile after 20 seconds. In Figure 4.2(a), the analytic solution (4.13) was used, and one can see that the velocity profiles match and indicate that the stationary state is kept in time. As a reference, Figure 4.2(b) shows a simulation where the initial state was not stationary. The velocity field was set to

$$u(y) = \frac{u_{\max}}{2}\left(1 - \frac{|y-h|}{h}\right), \tag{4.14}$$

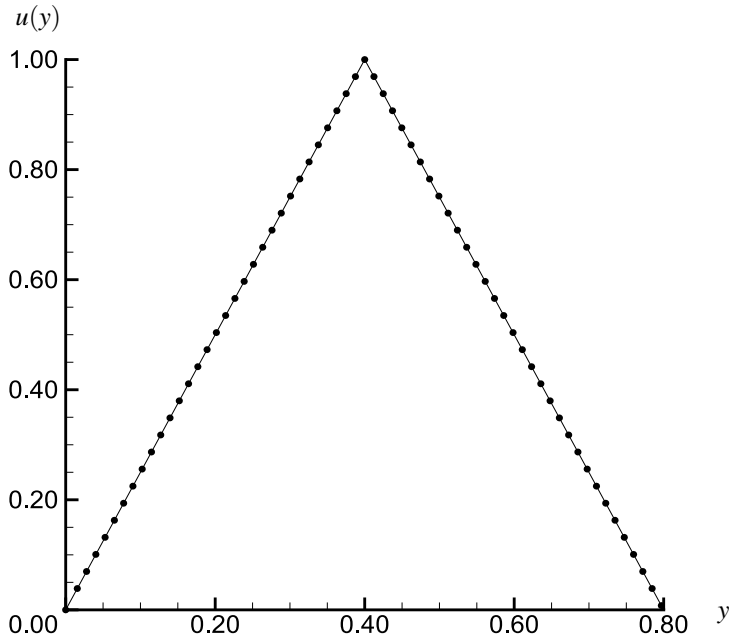
and as expected, the velocity profiles in the right-hand figure do not match.

4.2.2 A tilted interface

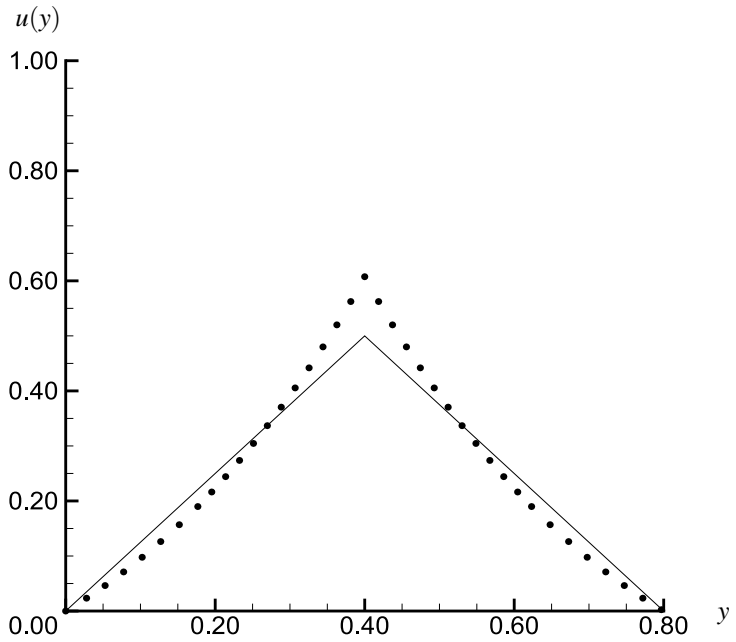
It has been verified that an orthogonal rotation of the axes gives the same results as the ones discussed above. As mentioned just before equation (4.11), all but one Marangoni term is zero for both the horizontal and the vertical case. To test the remaining six terms, a tilted case is considered where the system described in equations (4.4) to (4.12) is rotated 45 degrees counter-clockwise relative to the axes. This is a much more general case than the corresponding horizontal and vertical cases, because an arbitrary interface will normally not be aligned with the axes.

In the tilted case, the coefficient of surface tension becomes

$$\sigma(x,y) = \sigma_0 + \frac{\sigma_1}{\sqrt{2}}(x+y), \tag{4.15}$$



(a) Initial velocity set to the steady state velocity.



(b) Initial velocity set to half the steady state velocity.

Figure 4.2: Flow velocity profiles for two simulations. The solid line is the initial profile and the dotted line is the profile after 20 simulated seconds. In (a), the steady state-solution is given as initial condition, and in (b) a false steady-state solution is given as initial condition. The simulations show that the steady-state solution is kept in time, whereas the false solution is not.

Table 4.1: Right-hand side vector errors, $E(n_i)$.

n	$E(n_i)$	N
25	$2.471 \cdot 10^{-2}$	-
51	$1.211 \cdot 10^{-2}$	1.00
101	$6.115 \cdot 10^{-3}$	1.00
201	$3.073 \cdot 10^{-3}$	1.00
401	$1.549 \cdot 10^{-3}$	0.99
801	$7.711 \cdot 10^{-4}$	1.01

and the steady-state velocity-field becomes $\boldsymbol{u}(x,y) = \big(u(x,y), v(x,y)\big)$, where

$$u(x,y) = v(x,y) = \frac{h\sigma_1}{2\sqrt{2}\mu}\left(1 - \frac{|y-x|}{\sqrt{2}h}\right). \tag{4.16}$$

Because of the staggered grid, the velocities u and v are initialized as $u(x_e, y)$ and $v(x, y_n)$, where x_e is the value of x on the east edge, and y_n is the value of y on the north edge.

It is no longer possible to simulate an infinite tube using the implemented framework, so the analytic solution can not be tested in the same way as was shown for the horizontal case. It can still be tested, though, by noting that the analytic solution still holds everywhere but at the boundaries at the first time step. This means that the right-hand side vector, $\boldsymbol{a}$, defined in equation (3.3) should be zero everywhere except near the boundaries. Hence, the initial right-hand side vector is calculated on several grid sizes n, and the error $E(n)$ is computed for each n, where the error is defined as the maximum norm of $\boldsymbol{a}$, i.e.

$$E(n) = \|\boldsymbol{a}(n)\|_\infty. \tag{4.17}$$

Also, the convergence rate is calculated using

$$N_i = \log\left(\frac{E(n_i)}{E(n_{i-1})}\right) / \log\left(\frac{n_{i-1}}{n_i}\right). \tag{4.18}$$

Note that the values close to the boundaries are removed from $\boldsymbol{a}$ before $E(n)$ is calculated, since the forces do not balance there. Table 4.1 shows the errors $E(n)$ for several grid sizes n. The table shows that the convergence rate is one, which agrees with expectations since the GFM is a first-order method.

As a reference to the calculated errors, the error for the corresponding horizontal case when $n = 25$ is found to be $E(25) = 1.9 \cdot 10^{-15}$. The errors in the tilted case are thus far greater than the error in the horizontal case. To find the source of the difference in the error, the separate terms of the right-hand side vector, i.e. the convective and Laplacian terms and the body force, are calculated and compared. The body force is zero in both cases. In the horizontal case, the convective term has no contribution to the error, so the error in the Laplacian is $E(25) = 1.9 \cdot 10^{-15}$. In the tilted case, the error in the Laplacian was of order 10^{-15}, exposing the convective term as the source of the error difference. Table 4.2 shows the errors for the convective term for several grid sizes n, and shows that the error is of order $O(\Delta x)$. Note that the implementation of the Marangoni terms does not affect the convective term, so the small error in the Laplacian term indicates that the implementation is correct.

The explanation for the error in the convective term is that in the presence of a surface tension gradient, there is a sharp discontinuity in the derivative of the velocity at the interface. This can be

Table 4.2: Errors in the convective term for the tilted case.

n	Δx	Convective
25	$3.20 \cdot 10^{-2}$	$2.47 \cdot 10^{-2}$
51	$1.57 \cdot 10^{-2}$	$1.21 \cdot 10^{-2}$
101	$7.92 \cdot 10^{-3}$	$6.12 \cdot 10^{-3}$
201	$3.98 \cdot 10^{-3}$	$3.07 \cdot 10^{-3}$
401	$2.00 \cdot 10^{-3}$	$1.54 \cdot 10^{-3}$
801	$9.99 \cdot 10^{-4}$	$7.71 \cdot 10^{-4}$

seen from equations (2.43) to (2.46) by letting $[\mu] = 0$, resulting in

$$[\nabla \boldsymbol{u}] = \frac{\partial \sigma}{\partial x} \begin{bmatrix} -n_1 n_2^2 & -n_2^3 \\ n_1^2 n_2 & n_1 n_2^2 \end{bmatrix} + \frac{\partial \sigma}{\partial y} \begin{bmatrix} n_1^2 n_2 & n_1 n_2^2 \\ -n_1^3 & -n_1^2 n_2 \end{bmatrix}. \tag{4.19}$$

Hence, the convective term

$$\boldsymbol{u} \cdot \nabla \boldsymbol{u} = \begin{bmatrix} u u_x + v u_y \\ u v_x + v v_y \end{bmatrix}, \tag{4.20}$$

might also be discontinuous across the interface. The discretization of the convective term does not consider any such discontinuity, and might result in a smearing error close to the interface.

A calculation of equation (4.20) using the velocities given in equations (4.12) and (4.16) shows that the convective term is zero in both the horizontal and the tilted case, as should be expected. The convective term is thus continuous across the interface in both of these cases, even though $\nabla \boldsymbol{u}$ is not. In the tilted case where none of the components of the velocity or its derivative are zero, the discretization leads to numerical smearing of the discontinuity in $\nabla \boldsymbol{u}$ across the interface. Small errors therefore appear close to the interface. When the interface is aligned with the axes as in the horizontal case, only one of the four components of $\nabla \boldsymbol{u}$ is non-zero, i.e. $\partial u / \partial y$. The smearing of the derivative is therefore completely removed, since $v = 0$, and the convective term is correctly calculated as zero everywhere.

The plane interface tests have all indicated that the implementation of the Marangoni terms is correct. The simulations agreed with the analytic results, and it has been shown that the Marangoni effect induces a flow in the direction of increasing surface tension. The tilted test has exposed a weakness in the numerical method. The inclusion of Marangoni effect induces discontinuities in the derivative of the velocity across the interface that should be considered in the discretization of the convective term. This is not handled by the standard GFM. Li & Lai (2001) presents an extended version of the immersed interface method (IIM) which seems to account for the discontinuities in $\nabla \boldsymbol{u}$. The IIM, due to LeVeque & Li (1994), is much more involved than the GFM, so a straightforward adaptation is not recommended. A correction term based on Li & Lai (2001) might still be adapted into the method used in this work, and so further investigation along this line is suggested.

4.3 A droplet in a linear temperature field

Consider a two-dimensional droplet with radius r in an unbounded fluid medium, see Figure 4.3. Again, for simplicity, let the viscosity and density of the two fluids match, and let the surface tension vary linearly, e.g.

$$\sigma = \sigma_0 + \sigma_1 x. \tag{4.21}$$

Assume that the pressure outside the droplet is zero, and let both the droplet and the surrounding fluid be initially at rest such that $\boldsymbol{u} = \boldsymbol{0}$ everywhere.

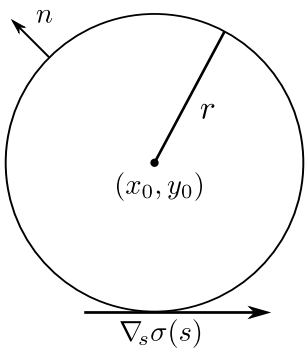


Figure 4.3: An idealized droplet with radius r, centred at (x_0, y_0).

4.3.1 Initial analysis

Let $\boldsymbol{F}_d$ and $\boldsymbol{F}_s$ denote the initial net forces on the droplet and the surrounding fluid, respectively. Since $\boldsymbol{u} = 0$, the only initial forces acting on the droplet are the net Marangoni force, $\boldsymbol{F}_M$, and the net capillary force, $\boldsymbol{F}_c$, i.e.

$$\boldsymbol{F}_d = \boldsymbol{F}_M + \boldsymbol{F}_c. \tag{4.22}$$

The fluid properties are equal, so the Marangoni force acts on the surrounding fluid with equal magnitude, that is

$$\boldsymbol{F}_s = \boldsymbol{F}_M. \tag{4.23}$$

The capillary force acting on the droplet must be matched by the pressure inside the droplet, such that

$$[p] = p^+ - p^- = -\frac{\sigma(x)}{r}, \tag{4.24}$$

where $^+$ and $^-$ denote the different fluids, see equation (2.38). Since the pressure in the surrounding fluid is assumed zero, $p^+ = 0$, the pressure inside the droplet must be

$$p^- = \frac{\sigma(x)}{r}. \tag{4.25}$$

Figure 4.4 shows several sketches of the initial forces in the system. Figure 4.4(a) shows the Marangoni forces, Figure 4.4(b) shows the capillary forces and Figure 4.4(c) shows the total surface tension forces, i.e. the sum of the Marangoni and the capillary forces. Figure 4.4(d) shows the net forces on the droplet and the surrounding medium. Note that the forces have the same magnitude and point of attack.

One of the intermediate results from Section 2.2 can be used to find an analytic expression for the net forces, $\boldsymbol{F}_d$ and $\boldsymbol{F}_s$. Consider the equation (2.25),

$$\oint_\Gamma \varphi \left([\boldsymbol{T}] \cdot \boldsymbol{n} + \boldsymbol{f}_{\text{sfd}} \right) \mathrm{d}s = 0, \tag{4.26}$$

where φ is an arbitrary, smooth function with compact support and $\boldsymbol{f}_{\text{sfd}} = \sigma\kappa\boldsymbol{n} + \boldsymbol{\nabla}_s\sigma$ is the surface force density. Let φ be chosen such that $\varphi = 1$ in the neighbourhood of the droplet, then equation (4.26) becomes

$$\oint_\Gamma [\boldsymbol{T}] \cdot \boldsymbol{n} \, \mathrm{d}s = -\oint_\Gamma (\sigma\kappa\boldsymbol{n} + \boldsymbol{\nabla}_s\sigma) \, \mathrm{d}s. \tag{4.27}$$

The left-hand side can be written as the sum of the net forces,

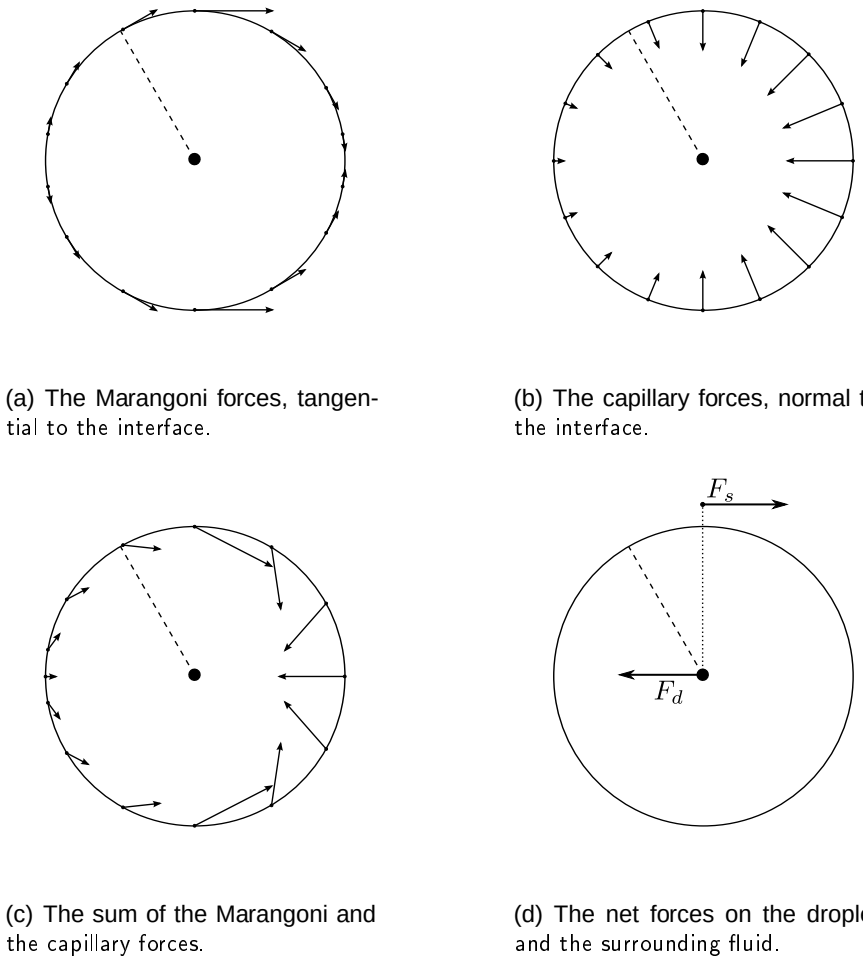
$$\begin{aligned}\oint_{\Gamma}[\boldsymbol{T}]\cdot\boldsymbol{n}\,\mathrm{d}s &= \oint_{\Gamma}\boldsymbol{T}^{+}\cdot\boldsymbol{n}\,\mathrm{d}s-\oint_{\Gamma}\boldsymbol{T}^{-}\cdot\boldsymbol{n}\,\mathrm{d}s\\ &= -\left(\oint_{-\Gamma}\boldsymbol{T}^{+}\cdot\boldsymbol{n}\,\mathrm{d}s+\oint_{\Gamma}\boldsymbol{T}^{-}\cdot\boldsymbol{n}\,\mathrm{d}s\right)\\ &= -\left(\boldsymbol{F}_s+\boldsymbol{F}_d\right),\\ &= -\left(\boldsymbol{F}_c+2\boldsymbol{F}_M\right).\end{aligned} \tag{4.28}$$

By comparing the left-hand side expression given in equation (4.28) with the right-hand side of equation (4.26), one obtains

$$\boldsymbol{F}_c=\oint_{\Gamma}\sigma\kappa\boldsymbol{n}\,\mathrm{d}s, \tag{4.29}$$

and

$$\boldsymbol{F}_M=\frac{1}{2}\oint_{\Gamma}\boldsymbol{\nabla}_s\sigma\,\mathrm{d}s. \tag{4.30}$$



(a) The Marangoni forces, tangential to the interface.

(b) The capillary forces, normal to the interface.

(c) The sum of the Marangoni and the capillary forces.

(d) The net forces on the droplet and the surrounding fluid.

Figure 4.4: The surface-tension forces on a droplet and the surrounding fluid due to a linear variation in surface tension. (d) shows the net forces on the droplet and the surrounding fluid. Both forces in (d) have the same point of attack.

The analytic expressions for equations (4.29) and (4.30) can be found by a calculation of the integrals. Assume for simplicity that $x_0 = y_0 = 0$. The droplet interface can be parametrized in Cartesian coordinates as two half circles,

$$y = \pm\sqrt{r^2 - x^2}, \qquad -r \leq x \leq r. \tag{4.31}$$

The unit normal, the curvature and the interface-element $\mathrm{d}s$ can be calculated as

$$\boldsymbol{n} = \frac{1}{r}\begin{bmatrix} x \\ \pm\sqrt{r^2 - x^2} \end{bmatrix}, \tag{4.32}$$

$$\kappa = -\boldsymbol{\nabla}\cdot\boldsymbol{n} = -\frac{1}{r} \tag{4.33}$$

and

$$\mathrm{d}s = \sqrt{1 + \big(y'(x)\big)^2}\,\mathrm{d}x = \frac{r}{\sqrt{r^2 - x^2}}\,\mathrm{d}x. \tag{4.34}$$

The normal and the curvature can then be used to find the integrand of equation (4.29), which becomes

$$\sigma\kappa\boldsymbol{n} = -\left(\frac{\sigma_0 + \sigma_1 x}{r^2}\right)\begin{bmatrix} x \\ \pm\sqrt{r^2 - x^2} \end{bmatrix}. \tag{4.35}$$

The integrand in equation (4.30) can be found using equation (2.10), giving

$$\boldsymbol{\nabla}_s\sigma = \frac{\sigma_1}{r^2}\begin{bmatrix} r^2 - x^2 \\ \pm x\sqrt{r^2 - x^2} \end{bmatrix}. \tag{4.36}$$

Now the Marangoni and capillary forces are calculated, resulting in

$$\boldsymbol{F}_c = -\pi\sigma_1 r\,\boldsymbol{i}, \tag{4.37}$$

and

$$\boldsymbol{F}_M = \frac{\pi\sigma_1 r}{2}\boldsymbol{i}, \tag{4.38}$$

where $\boldsymbol{i}$ is the unit vector in the x-direction. Hence, the initial net forces on the droplet and on the surrounding fluid are

$$\boldsymbol{F}_d = -\frac{\sigma_1\pi r}{2}\boldsymbol{i} \tag{4.39}$$

and

$$\boldsymbol{F}_s = \frac{\sigma_1\pi r}{2}\boldsymbol{i}. \tag{4.40}$$

Equations (4.39) and (4.40) shows that a surface tension that varies linearly in the x-direction drives the droplet to flow in the direction of decreasing surface tension. This can also be explained using the result from the previous section, where it was shown that a variation in the surface tension along an interface starts a flow along the interface in the direction of increasing surface tension. For a droplet where the surface tension varies linearly in the x-direction, this induced flow means that there must be two circulation currents inside the droplet and a net flow in the positive x-direction in the surrounding fluid. The droplet must then obtain a net velocity in the negative x-direction. If this was not so, the net force on the system would be non-zero, i.e.

$$\boldsymbol{F}_s + \boldsymbol{F}_d \neq 0, \tag{4.41}$$

which means that the initial pressure gradient outside the droplet would be non-zero.

Table 4.3: Calculations of the initial force on the droplet.

n	$\overline{\boldsymbol{F}}_d$ [N]	N
25	2.1943	-
50	2.3460	0.989
100	2.4217	0.976
200	2.4643	1.133
400	2.4799	0.829
800	2.4925	1.422

4.3.2 Numerical results

The droplet is initialized in a box with side lengths h. To simulate an infinite medium, the length h will be chosen for each set of simulations such that the boundaries do not affect the flow, typically $h \sim 20r$. All simulations will use the same fluid parameters:

$$\sigma_0 = \frac{0.1}{\pi}\ \mathrm{N/m}, \qquad \sigma_1 = -\frac{0.1}{\pi}\ \mathrm{N/m^2},$$
$$\rho = 1.0\ \mathrm{kg/m^2}, \qquad \mu = 0.005\ \mathrm{Pa\,s},$$
$$r = 0.005\ \mathrm{m}.$$

These parameters result in an initial force on the droplet in the positive x-direction with magnitude

$$|\boldsymbol{F}_d| = 2.5 \cdot 10^{-4}\,\mathrm{N}, \tag{4.42}$$

and the corresponding force on the surroundings is

$$\boldsymbol{F}_s = -\boldsymbol{F}_d. \tag{4.43}$$

The pressure inside the droplet at the points (x,y) is given by equation (4.25), which in this case becomes

$$p^-(x) = \frac{20}{\pi}(1-x) \simeq 6.37\,(1-x). \tag{4.44}$$

In the first numerical test, the velocity and pressure are initialized to zero, and the Poisson equation is solved once to update the pressure inside the droplet. The initial force on the droplet is then calculated using the identity

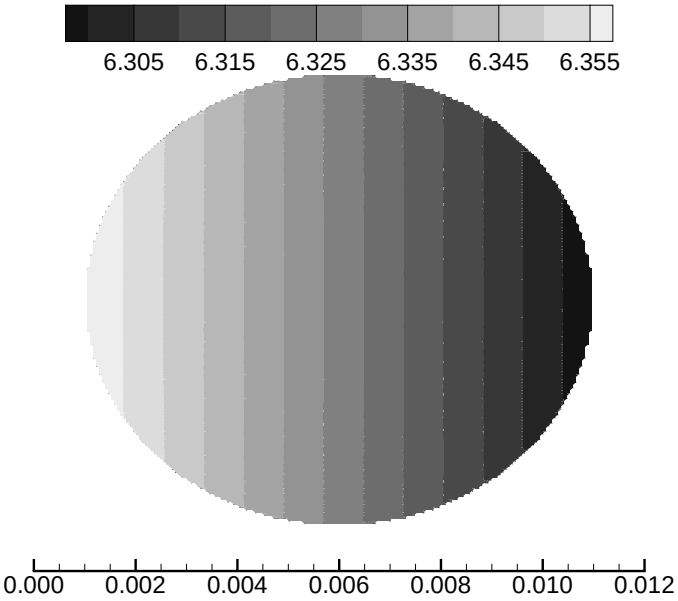
$$\boldsymbol{F}_d = \int_\Gamma \boldsymbol{T}\cdot\boldsymbol{n}\,\mathrm{d}s = \int_D \nabla\cdot\boldsymbol{T}\,\mathrm{d}V, \tag{4.45}$$

where D is the droplet domain, and the approximated force will be denoted $\overline{\boldsymbol{F}}_d$. The integral is calculated using a second-order numerical quadrature. Since the force is only considered initially when $t = 0$, the side-length h can be chosen close to the diameter of the droplet. In the following calculations, $h = 0.012\,\mathrm{m}$ is used.

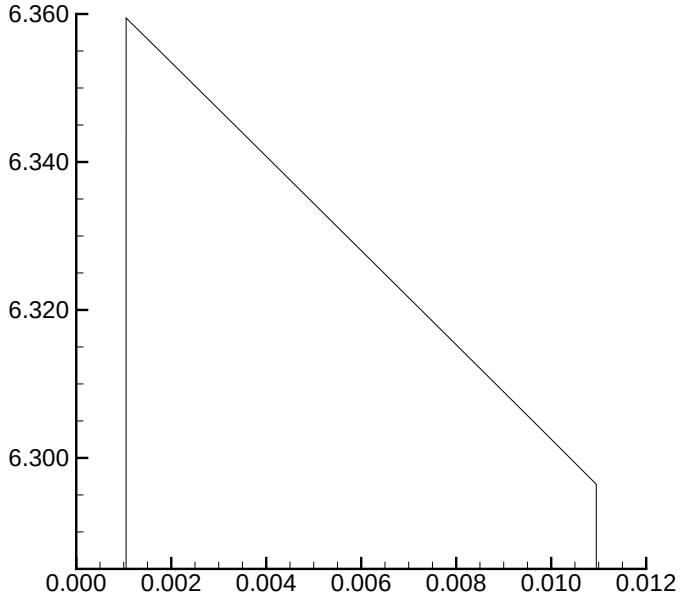
Figure 4.5 shows two plots of the pressure. Figure 4.5(a) shows a contour plot of the pressure, and Figure 4.5(b) shows the pressure profile along the middle of the droplet in the x-direction. The plots show that the calculated pressure is consistent with the analytic result given in equation (4.44).

The calculation results for the droplet force for different grid sizes $n \times n$ are given in Table 4.3. The convergence rates are calculated using the equation (4.18), where $E(n)$ is the absolute error of the calculated force $\overline{\boldsymbol{F}}_d(n)$ compared with the analytic result, i.e.

$$E(n) = |\overline{\boldsymbol{F}}_d(n) - \boldsymbol{F}_d|. \tag{4.46}$$



(a) A pressure contour-plot that shows a linear decrease in pressure along the x-axis.



(b) The pressure pro le along the centre of the droplet in the x-direction.

Figure 4.5: The initial pressure inside the droplet. The plots show that the initial pressure is consistent with equation (4.25).

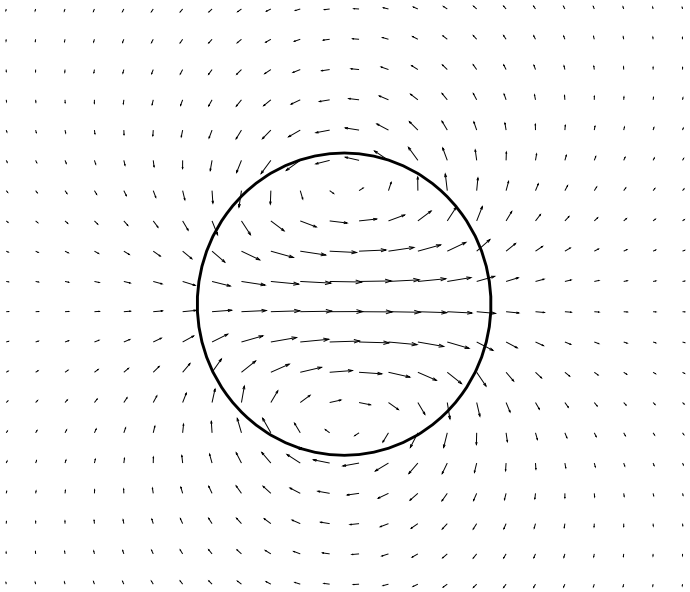
The results indicate a first-order convergence towards the analytic results. This is as expected, since the GFM is a first-order method. The same qualitative results have been obtained for a 60-degree rotation of the system counter-clockwise. The results show that the Marangoni force acts on the droplet in the direction of decreasing surface-tension, which is in accordance with literature, see e.g. Brennen (2005, Section 3.3).

Next, a simulation of the droplet is made using a rectangular box with vertical side-length $h_1 = 0.1$ m and horizontal side-length $h_2 = 0.12$ m. The droplet is placed in the centre of the box, and the simulation is run until $t = 0.07$ s. Figure 4.6(a) shows the velocity vector-field close to the droplet at $t = 0.07$ s. It can be seen that the velocity at the droplet interface is positive, and that there are internal and external currents. Figure 4.6(b) illustrates this by showing the velocity vector field relative to the droplet. The circulation currents inside the droplet can be seen, and the external flow is seen to be in the negative x-direction. This is as expected when compared to the discussion in the previous section.

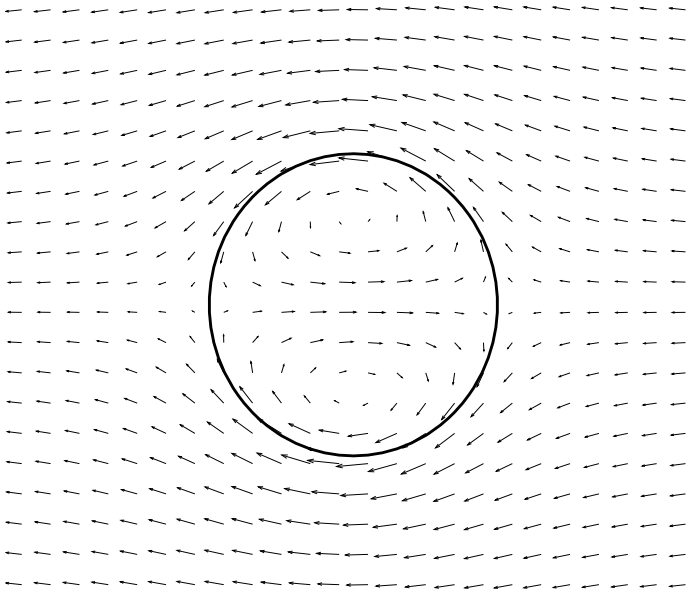
The analysis of the droplet has been restricted to a two-dimensional droplet. For an analytic treatment of the Marangoni force on a three-dimensional droplet, see Subramanian (1985). The results in this section indicate that the implementation is capable of simulating two-phase incompressible flow with a known variation in surface tension. The Marangoni effect has been demonstrated, and the results have been shown to be in accordance with literature.

4.4 Summary

The Marangoni effect has been studied, and two idealized cases have been used as examples. The cases showed that a variation in surface tension induces a flow in the direction of increasing surface tension. For a droplet, this was shown to result in a force in the direction of decreasing surface tension. The examples have demonstrated that the implementation is correct, and capable of handling two-phase flow with a given surface-tension variation.



(a) The velocity vector-field at $t = 0.07$s.



(b) The velocity vector-field relative to the droplet at $t = 0.07$s.

Figure 4.6: The velocity vector-fields close to the droplet at $t = 0.07$ s. The plots show that the Marangoni force induces a net flow in the negative x-direction outside the droplet, and two circulation currents inside the droplet.

5 Surfactants

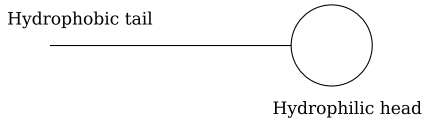
Hansen (2005, Chapter 9.2) stated that a natural extension to the two-phase modelling framework in order to describe more realistic systems, was to include a conservation law for surfactants along the interface. This will allow the coefficient of the surface tension to be a function of the surfactant concentration, and will lead to tangential forces along the interface as described in the previous chapter.

The goal of this chapter is to present a surfactant model that can be coupled with the existing model. The chapter is divided into two parts. The first part is an introduction to surfactants and their effects, while the second part gives a derivation and discussion of a general model.

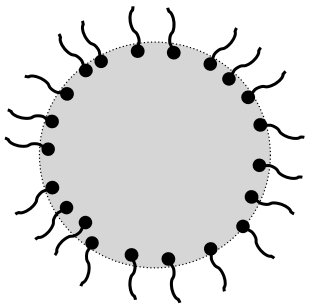
5.1 Introduction

Surfactants, or **surf**ace-**act**ive **a**ge**nts**, are amphipathic molecules, which means that they typically consist of a hydrophilic head and a long hydrophobic tail, see Figure 5.1(a). Common examples of surfactants are detergents, for example soap, but surfactants are also found in crude oil and on the coating of the surfaces of the alveoli, which are the small air sacks in the lungs serving as the site of exchange. Because of their amphipathic structure, surfactants tend to align themselves at the surface and internally so that the hydrophile end is toward the water and the hydrophobic tail is squeezed away from the water, see Figure 5.1(b) (Zumdahl, 2002). There are several types of surfactants, some of which are insoluble and others of which are soluble. Insoluble surfactants remain on the surface, whereas soluble surfactants diffuse into the bulk of the suspending fluid, see Bush (2004).

The presence of surfactants on the interface reduces the local surface tension, essentially because the molecules take up a preferred orientation in the surface layer, and exert on each other a repulsive force which partially balances the tension of the pure water surface, see (Batchelor, 1967, Section 1.9). It follows that gradients in surfactant concentration result in surface-tension gradients and the Marangoni effect. On the other hand, the surfactant concentration is affected by the motion of the surrounding fluid: The concentration of surfactant is both advected by the



(a) A schematic sketch of a surfactant molecule.



(b) Surfactants adsorbed on a water-oil interface.

Figure 5.1: Sketches of surfactants.

ambient flow and diffused along the moving interface, see Xu & Zhau (2003). Hence the motion of the fluid and the surfactant distribution are coupled together. This coupling requires a conservation law for surfactants along the interface and an equation of state. In the most general case, the conservation law must include both an advection and a diffusion term, and a source term associated with adsorption onto or desorption from the surface.

Xu *et al.* (2006) presents a level-set method for the simulation of fluid interfaces with insoluble surfactants in two-dimensions. The method couples a surfactant model based on Xu & Zhau (2003) with the immersed interface method developed by LeVeque & Li (1994) for solving the fluid flow equations. In James & Lowengrub (2004), an axisymmetric numerical method for simulating the dynamics of insoluble surfactants on a moving liquid-liquid interface is presented. In the following section, a surfactant model based on Xu & Zhau (2003) will be reviewed. The model is restricted to insoluble surfactants, where there is no transfer from or to the bulk either to or from the interface.

5.2 A surfactant model

Let f denote the surfactant concentration along the surface. The surface tension can be related to the surfactant concentration by the Langmuir equation of state, which expresses the coefficient of surface tension, σ, as a function of surfactant concentration, f, i.e.

$$\sigma(f) = \sigma_0 + RTf_\infty \log(1 - f/f_\infty). \tag{5.1}$$

Here σ_0 is the coefficient of surface tension for a clean interface where $f = 0$, R is the ideal gas constant, T is the temperature and f_∞ is the surfactant concentration at maximum packing. If the surfactant concentration $f \ll f_\infty$, then the approximation

$$\log(1 - f/f_\infty) \simeq -f/f_\infty \tag{5.2}$$

can be used to simplify equation (5.1), resulting in the linear approximation

$$\sigma(f) = \sigma_0 - RTf. \tag{5.3}$$

Further details about the equation of state and the modelling of surfactant interactions can be found in Pawar & Stebe (1996).

The conservation law for surfactants can be derived by considering a mass balance on an arbitrary surface element on the interface. Following is a short derivation based on Stone (1990) for two-dimensional deforming surfaces. Let $m(t)$ denote an arbitrary surface element, and consider a mass balance on $m(t)$. In the absence of diffusion, chemical reactions and fluxes to the surface from either of the surrounding bulk phases, the mass balance yields

$$\frac{\mathrm{D}}{\mathrm{D}t} \int_{m(t)} f \, \mathrm{d}m = 0, \tag{5.4}$$

where $\mathrm{D}/\mathrm{D}t$ denotes the convective derivative. Equation (5.4) can be rewritten by using the product rule, giving

$$\int_{m(t)} \left(\frac{\mathrm{D}f}{\mathrm{D}t} \, \mathrm{d}m + f \frac{\mathrm{D}}{\mathrm{D}t} \, \mathrm{d}m \right) = 0, \tag{5.5}$$

where $f\,\mathrm{D}\,\mathrm{d}m/\mathrm{D}t$ accounts for the change in the material surface element as a result of stretching and distortion. Define the vector area $\mathrm{d}\boldsymbol{m} = \boldsymbol{n}\,\mathrm{d}m$, where $\boldsymbol{n}$ is the unit normal vector. It can be shown that the time rate of change of a material surface element with a vector area $\mathrm{d}\boldsymbol{m}$ is,

$$\frac{\mathrm{D}}{\mathrm{D}t} \mathrm{d}\boldsymbol{m} = \mathrm{d}\boldsymbol{m} \boldsymbol{\nabla} \cdot \boldsymbol{u}_I - (\boldsymbol{\nabla} \boldsymbol{u}_I) \cdot \mathrm{d}\boldsymbol{m}, \tag{5.6}$$

where $\boldsymbol{u}_I$ is the local interface velocity, see Batchelor (1967, page 132). Since the fluid velocity is continuous across the interface, $\boldsymbol{u}_I = \boldsymbol{u}$. Taking the inner product of equation (5.6) with $\boldsymbol{n}$ gives

$$\frac{\mathrm{D}}{\mathrm{D}t}\,\mathrm{d}m = \boldsymbol{\nabla}_\mathrm{s}\cdot\boldsymbol{u}\,\mathrm{d}m, \tag{5.7}$$

where $\boldsymbol{\nabla}_\mathrm{s}$ is the surface gradient operator, see equation (2.10). Since f is only defined along the interface, the convective derivative of f in equation (5.5) is

$$\frac{\mathrm{D}f}{\mathrm{D}t} = \frac{\partial f}{\partial t} + \boldsymbol{u}_\mathrm{s}\cdot\boldsymbol{\nabla}_s f, \tag{5.8}$$

where $\boldsymbol{u}_\mathrm{s}$ represents the tangential component of the velocity, $\boldsymbol{u}_\mathrm{s} = (\mathbf{1} - \boldsymbol{nn})\,\boldsymbol{u}$. Combining equations (5.5), (5.7) and (5.8) gives

$$\int_{m(t)} \left(\frac{\partial f}{\partial t} + \boldsymbol{u}_\mathrm{s}\cdot\boldsymbol{\nabla}_s f + \boldsymbol{\nabla}_\mathrm{s}\cdot\boldsymbol{u} f \right) \mathrm{d}m = 0, \tag{5.9}$$

and since $m(t)$ is arbitrary, it follows that

$$\frac{\partial f}{\partial t} + \boldsymbol{\nabla}_\mathrm{s}\cdot(f\boldsymbol{u}) = 0. \tag{5.10}$$

If the velocity $\boldsymbol{u}$ is decomposed into the tangential component $\boldsymbol{u}_\mathrm{s}$ and the normal component $\boldsymbol{u}_\mathrm{n} = \boldsymbol{u} - \boldsymbol{u}_\mathrm{s}$, equation (5.10) can be rewritten as

$$\frac{\partial f}{\partial t} + \boldsymbol{\nabla}_\mathrm{s}\cdot(f\boldsymbol{u}_\mathrm{s}) + (\boldsymbol{u}\cdot\boldsymbol{n})\kappa f = 0, \tag{5.11}$$

where κ is the mean curvature, see Xu & Zhau (2003). Here the second term corresponds to the tangential advection along the interface, and the third term corresponds to surface stretching by the normal velocity. James & Lowengrub (2004, Appendix A) shows that equation (5.11) can be written in the following equivalent formulation:

$$\frac{\partial f}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla} f - \boldsymbol{n}\cdot\boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{n} f = 0, \tag{5.12}$$

where $\boldsymbol{n}\cdot\boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{n}$ is the normal component of the normal derivative. The surface diffusion is included in the above equation by adding a diffusion term, $D_\mathrm{s}\boldsymbol{\nabla}_\mathrm{s}^2 f$. This results in the convective-diffusion equation for surfactant transport,

$$\frac{\partial f}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla} f - \boldsymbol{n}\cdot\boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{n} f = D_\mathrm{s}\boldsymbol{\nabla}_\mathrm{s}^2 f, \tag{5.13}$$

where D_s is the surface diffusivity of surfactant, see e.g. James & Lowengrub (2004) or Xu & Zhau (2003).

To capture the evolution of the surfactant concentration on the interface, Xu & Zhau (2003) suggests to use an approach similar to the level-set formulation. That is, define an auxiliary function $\hat{f}(\boldsymbol{x},t)$ such that for $\boldsymbol{x}\in\Gamma$, $\hat{f}(\boldsymbol{x},t) = f(\boldsymbol{x},t)$. The function $\hat{f}$ can be defined arbitrarily as long as it agrees with f on the interface. Xu & Zhau (2003) states that the best extension of f off the interface is the orthogonal extension, i.e. $\boldsymbol{\nabla}\hat{f}\cdot\boldsymbol{\nabla}\phi = 0$. The extension can be obtained using a surfactant-extrapolation equation similar to the velocity-extrapolation equation (3.13),

$$\frac{\partial \boldsymbol{f}}{\partial \tau} + S(\phi_0)\boldsymbol{n}\cdot\boldsymbol{\nabla} f = 0, \tag{5.14}$$

where the ˆ-notation for f off the interface is dropped. Just as the level-set function needs to be reinitialized, orthogonality between f and ϕ must be maintained. This is done by solving equation (5.14) at regular intervals.

A solution strategy is proposed in Xu & Zhau (2003). The outline of the main procedures that are proposed per time step is:

1. Evolve the surfactant concentration by solving equation (5.13). A semi-implicit modified Crank-Nicholson scheme of second-order accuracy in time is proposed for the time discretization, and a combination of second-order central differences and a third-order upwind scheme is proposed for the spatial discretization. It is shown that the proposed discretization yields second-order accuracy under the CFL condition $\Delta t = O(\Delta x)$.
2. Evolve the interface using the level-set method, see Section 3.2.
3. If needed, reinitialize the level-set function and extend the surfactant concentration off the interface orthogonal to ϕ.
4. Use the updated interface and surfactant concentration to update the velocity field.

5.3 Summary

This chapter has reviewed a model for surfactant conservation on an interface and surfactant influence on the surface tension. The model has been discussed, and a solution strategy proposed by Xu & Zhau (2003) has been outlined.

6 Conclusions and recommendations for further work

6.1 Conclusion

This thesis has treated a two-phase flow model for inviscid and Newtonian fluids with constant density and viscosity. The surface-tension force on the interface and the difference in fluid properties induce jumps in the pressure and the velocity and their derivatives across the interface. These jumps are handled by a set of jump conditions, which have been derived for a general surface-tension force.

It has been shown how a projection method is used to discretize the governing equations. The spatial discretization has been reviewed, and the ghost-fluid method for handling the interface discontinuities has been discussed. The level-set method is used to capture the interface, and SSP RK schemes are used for the temporal discretization.

The main contribution in this thesis is the extension of a two-phase flow model to include the effects of varying surface-tension along an interface separating two fluids. The jump conditions of the old framework have been modified, and two test cases have been studied. These test cases are used to validate the implementation as well as study the effects of Marangoni forces. It has been shown that a surface-tension gradient induces a flow along the interface in the direction of increasing surface tension. For a droplet, this was shown to result in a net force in the direction of decreasing surface tension. The tests indicate that the implementation of the Marangoni terms are correct, and that the model is capable of simulating two-phase flow with a given surface-tension variation. The tests also revealed a weakness in the standard GFM: The presence of Marangoni forces induce a discontinuity in the gradient of the velocity across the interface. Since this discontinuity is not explicitly accounted for in the discretization of the convective term, an error due to numerical smearing occurs.

Finally, the thesis has discussed surfactant effects on two-phase flow. Surfactants are explained and described, and a general surfactant-model is reviewed. The model includes an equation of state for coupling the surface tension with the surfactant concentration, and a transport equation for evolving the surfactant concentration in time. The model is discussed, and some implementation ideas are given.

6.2 Recommendations for further work

It is recommended that further work is made to implement and analyse a surfactant model. The model reviewed in this thesis can be used as a basis. The implementation will require a discretization of the transport equation and an implementation of an equation of state. If soluble surfactants are desirable, a transport equation is also needed for surfactant transport between the bulk and the interface.

It is also recommended that one finds a way to handle the interfacial discontinuity in the gradient of the velocity for the convective term. One suggestion is to adapt the correction term for the convective term given in Li & Lai (2001). This might not be straightforward, especially considering the higher-order methods, e.g. the ENO and WENO schemes.

Bibliography

Aris, R. (1989). *Vectors, Tensors and the Basic Equations of Fluid Mechanics*. Dover publications.

Batchelor, G. K. (1967). *An Introduction to Fluid Dynamics*. Cambridge University Press.

Brennen, C. E. (2005). *Fundamentals of Multiphase Flow*. Cambridge University Press, first ed.

Burden, R. L. & Faires, J. D. (2004). *Numerical Analysis*. Brooks Cole, eight ed.

Bush, J. W. M. (2004). Surface tension module. URL `http://web.mit.edu/1.63/www/Lec-notes/Surfacetension/`.

Chorin, A. J. & Marsden, J. E. (2000). *A Mathematical Introduction to Fluid Mechanics*. Springer.

Fedkiw, R. P. (1999). A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, *152*, 457–492.

Ferracina, L. & Spijker, M. N. (2004). Stepsize retrictions for the total-variation-diminishing property in general runge-kutta methods. *SIAM Journal of Numerical Analysis*, *42*, 1073–1093.

Ferracina, L. & Spijker, M. N. (2005). An extension and analysis of the shu-osher representation of runge-kutta methods. *Mathematics of Computation*, *239*, 201–219.

Fosse, I. (2006). *Simulations of a Falling Droplet and Two Approaching Droplets in a Newtonian Fluid*. Master's thesis, NTNU, Trondheim, Norway.

Ghia, U., Ghia, K. N., & Shin, C. T. (1982). High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *Journal of Computational Physics*, *48*(3), 387–411.

Griebel, M., Dornseifer, T., & Neunhoeffer, T. (1998). *Numerical Simulation in Fluid Dynamics*. SIAM.

Hansen, E. B. (2005). *Numerical Simulation of Droplet Dynamics in the Presence of an Electric Field*. Ph.D. thesis, NTNU, Trondheim, Norway.

Iserles, A. (1996). *A First Course in the Numerical Analysis of Differential Equations*. Cambridge University Press.

James, A. J. & Lowengrub, J. (2004). A surfactant-conserving volume-of-fluid method for interfacial flows with insoluble surfactant. *Journal of Computational Physics*, *201*(2), 685–722.

Jiang, G.-S. & Peng, D. (2000). Weighted eno schemes for hamilton jacobi equations. *Journal of Scientific Computing*, *21*, 2126–2143.

Jiang, G.-S. & Shu, C.-W. (1996). Efficient implementation of weighted eno schemes. *Journal of Computational Physics*, *126*, 202–228.

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

Ketcheson, D. I. & Robinson, A. C. (2005). On the practical importance of the ssp property for runge-kutta time integrators for some common godunov-type schemes. *International Journal for Numerical Methods in Fluids*, *48*, 271–303.

Landau, L. D. & Lifshitz, E. M. (1987). *Fluid Mechanics*. Oxford, Pergamon, second ed.

Langtangen, H. P., Mardal, K.-A., & Winther, R. (2002). Numerical methods for incompressible viscous flow. *Advances in Water Resources*, *25*(8), 1125–1146.

LeVeque, R. & Li, Z. (1994). The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal of Numerical Analysis*, *31*, 1019–1044.

Li, Z. & Lai, M.-C. (2001). The immersed interface method for the navier-stokes equations with singular forces. *Journal of Computational Physics*, *171*, 822–842.

Lundgaard (2005). Compact separation by electrocoalescence. URL `http://www.sintef.no/content/page1____3326.aspx`.

Osher, S. & Fedkiw, R. P. (2003). *The Level-Set Method and Dynamic Implicit Surfaces*. Springer.

Pawar, Y. & Stebe, K. J. (1996). Marangoni effects on drop deformation in an extensional flow: The role of surfactant physical chemistry. i. insoluble surfactants. *Physics of Fluids*, *8*(7), 1738–1751.

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. SIAM, second ed.

Sethian, J. A. (1999). *Level Set Methods and Fast Marching Methods*. Cambridge University Press.

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

Stone, H. A. (1990). A simple derivation of the time-dependent convective-diffusion equation for surfactant transport along a deforming interface. *Physics of Fluids*, *2*(1), 111–112.

Subramanian, R. S. (1985). The stokes force on a droplet in an unbounded fluid medium due to capillary effects. *Journal of Fluid Mechanics*, *153*, 389–400.

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

Teigen, K. E. (2007). *Numerical investigation of flow phenomena in two-phase flow of natural gas*. Master's thesis, NTNU, Trondheim, Norway.

Temam, R. (1995). *Navier-Stokes Equations and Nonlinear Functional Analysis*. SIAM, Philadelphia, second ed.

White, F. M. (2003). *Fluid Mechanics*. McGraw-Hill, fifth ed.

Xu, J.-J., Li, Z., Lowengrub, J., & Zhau, H. (2006). A level set method for interfacial flows with surfactants. *Journal of Computational Physics*, *212*(2), 590–616.

Xu, J.-J. & Zhau, H. (2003). An eulerian formulation for solving partial differential equations along a moving interface. *Journal of Scientific Computing*, *19*(1-3).

Zhao, H. K., Chan, T., Merriman, B., & Osher, S. (1996). A variational level set approach to multiphase motion. *Journal of Computational Physics*, *127*, 179–195.

Zumdahl, S. S. (2002). *Chemical Principles*. Houghton Mifflin College Div, fourth ed.