

Numerical Modelling of Two-Phase Flow with Moving Boundary Fitted Meshes

THÈSE N° 8538 (2018)

PRÉSENTÉE LE 7 JUIN 2018

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE TRANSFERT DE CHALEUR ET DE MASSE
PROGRAMME DOCTORAL EN ENERGIE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Erik GROS

acceptée sur proposition du jury:

Prof. K. A. J. Mulleners, présidente du jury
Prof. J. R. Thome, Prof. G. Rabello Dos Anjos, directeurs de thèse
Prof. A. Soldati, rapporteur
Dr Y. Sato, rapporteur
Prof. F. Gallaire, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2018

Acknowledgements

I would like to express my gratitude to my supervisors Prof John Thome and Prof Gustavo Anjos for giving me this opportunity to do research and for their kind and patient leadership style. My sincere thanks go to the members of my thesis jury: Prof Mulleners, Prof Soldati, Prof Gallaire and Dr Sato for their valuable time reading and commenting on this manuscript. I would like to thank all EPFL professors whose courses I attended for all the things they taught me. The European Union funded THERMAPOWER project (Thermal Management of High Power Microsystems using Multiphase Flows) is gratefully acknowledged for supporting this work by providing travel money for two research visits to the State University of Rio de Janeiro. In the last three years, while being a member of the LTCM lab, I enjoyed the company of PhD-students and postdocs from different nationalities. I am thankful to all my colleagues and friends for the great time spent together. Finally, I would like to thank my parents for their inexhaustible affection and support.

Lausanne, 11 May 2018

Erik Gros

Abstract

In this thesis, a computational approach is used to study two-phase flow including phase change by direct numerical simulation. This approach follows the interface with an adaptive moving mesh. The incompressible Navier-Stokes equations are solved, in two-dimensional and axisymmetric geometries, using the Finite Element Method (FEM). The computational domain is discretized using an unstructured triangular mesh and the mini element is used to satisfy the "inf-sup" compatibility condition. A combination of smoothing mesh velocities and remeshing is used to preserve the mesh quality. Adaptive mesh refinement is used to keep the mesh sufficiently refined where needed. Mesh adaptation strategies, allowing to control the refinement of the computational mesh, are discussed in the context of specific applications. The accurate representation of the interface between the phases is a key issue to model surface tension dominated flows. Here, the interface is represented explicitly by nodes and segments that are a subset of the computational mesh and a sharp transition of the fluid properties can be achieved. The surface tension force is included as a singular volume force, like in the continuum model (CSF). The present discretization is shown to allow for exact equilibrium (up to rounding errors) between the pressure and surface tension terms. This is important in order to suppress spurious currents, which are a common issue in computational two-phase flow. However, an exact computation of the interface curvature is necessary for the spurious currents to be numerically zero. The curvature of the interface, is efficiently and accurately computed by using the Frenet-Serret formulas. A phase change model is implemented via a source term in the continuity equation, which is computed from the jump in conductive heat flux at the interface. The presented approach is shown to provide an accurate description of different two-phase flow phenomena, including phase change, and to handle cases with large material property ratios. Accuracy and robustness of the present method are demonstrated on several benchmark cases, where the results are compared to analytical or semi-analytical solutions and experimental data.

Key words: Two-Phase Flow, Finite Element Method (FEM), Moving Mesh, Arbitrary Lagrangian Eulerian (ALE), Surface Tension, Phase Change.

Résumé

Dans cette thèse, une approche computationnelle est utilisée pour étudier le flux diphasique, y compris le changement de phase, par simulation numérique directe. Cette approche suit l'interface avec un maillage mobile adaptatif. Les équations de Navier-Stokes incompressibles sont résolues, en géométries bidimensionnelles et axisymétriques, en utilisant la méthode des éléments finis. Le domaine de calcul est discrétisé en utilisant un maillage triangulaire non structuré et l'élément mini est utilisé pour satisfaire la condition de compatibilité "inf-sup". Une combinaison de vitesses de maillage et de remaillage est utilisée pour préserver la qualité du maillage. Un raffinement de maillage adaptatif est utilisé pour maintenir le maillage suffisamment affiné. Les stratégies d'adaptation de maillage, permettant de contrôler le raffinement du maillage de calcul, sont discutées dans le contexte d'applications spécifiques. La représentation précise de l'interface entre les phases est un problème clé pour modéliser les écoulements dominés par la tension superficielle. Ici, l'interface est représentée explicitement par des points et des segments faisant partie du maillage et une transition nette des propriétés du fluide peut être obtenue. La force de tension de surface est incluse en tant que force de volume singulière à la manière du modèle CSF. Il est démontré qu'un équilibre exact (jusqu'aux erreurs d'arrondissement) est possible entre le gradient de pression et la tension de surface. Ceci est important afin de supprimer les courants parasites, qui sont un problème commun dans le cadre du calcul de flux diphasique. Cependant, un calcul exact de la courbure de l'interface est nécessaire pour que les courants parasites soient éliminés. La courbure de l'interface est calculée de manière efficace et précise en utilisant les formules de Frenet-Serret. Un modèle de changement de phase est implémenté via un terme source, qui est similaire au modèle de tension de surface et est calculé à partir du saut de flux de chaleur local à l'interface. Il est démontré que l'approche présentée peut fournir une description précise des forces interfaciales, du transfert de chaleur et de masse entre les phases et peut traiter les cas avec des rapports élevés de propriétés entre les deux phases. L'exactitude et la fiabilité de l'approche présentée sont démontrées pour plusieurs cas, où les résultats sont comparés à des solutions analytiques ou semi-analytiques et à des données expérimentales.

Mots clefs : écoulement biphasique, méthode des éléments finis, adaptation de maillage, Arbitrary Lagrangian Eulerian (ALE), tension de surface, changement de phase.

Contents

Acknowledgements	3
Abstract	5
1 Introduction	1
1.1 Capillary and Wetting Phenomena	2
1.1.1 Contact Lines	4
1.2 Finite Element Method	5
1.3 Motivation and Structure of the Thesis	6
2 Numerical Modelling of Two-Phase Flows: An Overview	9
2.1 Meso-Scopic Models	9
2.2 Macroscopic Models	11
2.2.1 Interface Tracking	12
2.2.2 Phase Change	16
3 Governing Equations	19
3.1 Mass and Momentum Equations	19
3.1.1 Two Fluids Formulation	20
3.1.2 One Fluid Formulation	22
3.2 Coordinate Systems	24
3.2.1 Axisymmetric Curvature	25
3.3 ALE Formulation	26
3.4 Conservation of Energy	27
3.5 Phase Change	29
4 Numerical Method	33

Contents

4.1	Finite Element Method	33
4.1.1	Variational Formulation of the Governing Equations	34
4.1.2	Semi Discrete System	36
4.1.3	LBB Compatibility Condition	37
4.1.4	Element Shape Functions	38
4.2	Surface Tension Force	39
4.3	Contact Angle	42
4.4	Phase Change	43
4.5	Semi-Lagrangian Time Discretization	44
4.6	Fully Discrete Equations	46
4.6.1	Algebraic Splitting	47
4.7	Boundary Conditions	47
4.7.1	Interface Boundary Conditions	49
4.8	Extension of Interface Quantities	49
4.9	Computational Mesh	50
4.9.1	Mesh Displacement	50
4.9.2	Mesh Refinement and Remeshing	52
4.10	Interpolation	53
4.11	Solution Algorithm	53
5	Computational Studies	57
5.1	Flow around a Sphere	57
5.1.1	Stokes Flow around a Sphere	57
5.1.2	Flow around Sphere for $Re = 200$	58
5.2	Natural Convection in a Heated Cavity	60
5.3	Annular Poiseuille Flow	61
5.4	Static Drop without Spurious Currents	64
5.5	Oscillating Droplet	66
5.6	Rising Bubble	67
5.7	2D Rayleigh-Taylor Instability	69
5.8	Axisymmetric Rayleigh-Taylor Instability	71
5.9	Travelling Waves in Liquid Film Falling on the Surface of a Fiber	74
5.10	Microchannel Simulations	77

5.11 Sessile Drop with static Contact Angle	80
5.12 1D Phase Change Problem	84
5.13 Evaporating Bubble in Superheated Liquid	86
5.14 Rayleigh-Taylor Instability with Phase Change	88
6 Conclusions and Outlook	95
6.1 Perspectives	96
6.1.1 Improvements to the Numerical Method	97
A Coordinate Transformations of the Navier-Stokes Equations	99
A.1 From Cartesian to Curvilinear Coordinates	99
A.2 Cylindrical Coordinates	101
Bibliography	103
Curriculum Vitae	115

1 Introduction

Flows with multiple phases are ubiquitous in nature and industrial applications. Multiphase flow is encountered every day in drinking, cooking, swimming and rain. Many geophysical phenomena and biological systems feature two or more phases, some examples are: sedimentation, water waves, cloud formation, land and snow slides, blood flow and temperature control by perspiration. There is a wide variety of engineering systems based on multiphase flow, such as: power plants, fuel injection in combustion engines [42], inkjet printing, lab-on-a-chip devices, distillation units, fluidized beds, chemical reactors, oil recovery and transportation [102], heat exchangers, evaporators, condensers, air conditioning, refrigeration and electronics cooling systems [91]. A sound understanding of multiphase flow is crucial for safe and efficient operation of these systems but the dynamics of multiphase flow can give rise to many complex phenomena. All the systems, listed above are governed by essentially the same physical laws describing mass, momentum and energy transport [71]. However, advances in engineering technology have required solutions to problems for which a purely theoretical approach is not practical. This is often achieved by using correlations and statistical or empirical approaches that have a limited range of validity. In the mean time, the rapid advent of computing power and the increased efficiency of numerical algorithms have allowed to simulate systems of increasing size and complexity. Numerical simulations offer detailed information of the flow field and provide a huge cost saving potential relative to experiments. This thesis is dedicated to a computational method for numerical simulation of two-phase flow.

Fluid flow can be the origin of, or produced by, different types of forces. Some examples of forces are gravity, viscosity, inertia and capillarity. The different forces have different scaling behaviours. For example, gravitational and other volume forces increase with the volume or the third power of the spatial size. Surface tension on the other hand, which is the origin of capillarity, is a surface phenomenon. As a result, when decreasing the spatial size the surface to volume ratio will increase and capillary forces will start to dominate over volume forces, which explains for instance the nearly spherical shape of small droplets and bubbles. Depending on the Reynolds number Re , which measures the ratio of inertia to viscous forces, the flow might

be turbulent or laminar. In this work we concentrate on flow in small geometries where the Reynolds number remains sufficiently small such that the flow is laminar. However, even at vanishing Reynolds number multiphase flow may exhibit non-linear effects because of the coupling between the position of the free boundaries or interfaces with the flow field. This non-linear nature of multiphase flows manifests itself by the large variety of flow patterns that can be observed in multiphase flow. Some two-phase flow patterns found in channels are illustrated in Fig. 1.1.

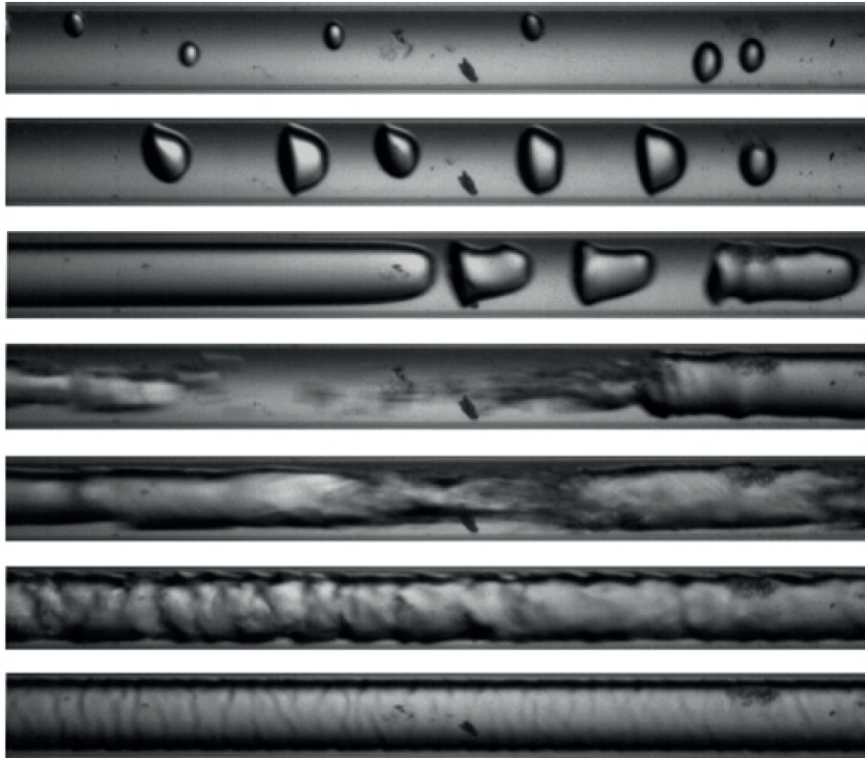


Figure 1.1: Two-phase flow patterns in a channel [111].

1.1 Capillary and Wetting Phenomena

Molecules at a liquid-gas or liquid-liquid interface have less interacting neighbors than molecules in the liquid bulk. This is the origin of the phenomenon called surface tension, due to which energy must be supplied to create interfaces and liquids adjust themselves to expose the smallest possible surface area [33]. Surface tension makes soap bubbles spherical and allows insects to walk on water. Moreover, it dictates the dynamics of two-phase flow through small geometries. Intermolecular interactions are described by Lennard-Jones-type potentials, which preconize that the (short range) repelling forces diminish faster with distance than the (long range) attracting forces [24]. Due to the density variation across a liquid-gas interface this leads to a reduced mean intermolecular distance normal to the interface compared to

the mean distance along the interface, since otherwise there would be a net force towards the liquid. Molecules on the interface therefore feel an increased attracting force from their neighbors on the interface. This results in a net tension force, felt by any molecule in the interface region, acting equally in all directions parallel to the interface.

Despite its molecular origin surface tension also has a precise macroscopic interpretation. It is a mechanical line force distribution, i.e. a force per unit length, pulling on any surface element of the interface from every side. Using some results from differential geometry, the surface tension force per unit interfacial area can be shown to be described by the following expression [18, 104]

$$\mathbf{f} = -\sigma \mathbf{n} (\nabla_S \cdot \mathbf{n}) + \nabla_S \sigma, \quad (1.1)$$

where σ is the surface tension coefficient, a property determined by the nature of the intermolecular forces of the two materials meeting at the interface, which is measured in N/m . In Eq. (1.1), use has been made of the interface gradient ∇_S defined as

$$\nabla_S = \nabla - \mathbf{n} (\mathbf{n} \cdot \nabla) = (\mathbf{I} - \mathbf{n}\mathbf{n}) \nabla, \quad (1.2)$$

where \mathbf{n} is the interface unit normal vector \mathbf{I} is the identity matrix and $\mathbf{n}\mathbf{n}$ is a dyadic product. That is, ∇_S represents the component of the gradient tangential to the interface and it acts on the interface in a similar way as ∇ acts on the whole space. The first term in Eq. (1.1) is the normal component of the surface tension force, while the second term is the tangential component. When the surface tension coefficient σ is constant, the tangential component vanishes and the surface tension force acts normal to the interface. The tangential component of surface tension is necessary to describe phenomena like Marangoni convection and thermo-capillary migration of drops. These phenomena are outside the scope of this work where σ is assumed to be a constant parameter.

The expression $-\nabla_S \cdot \mathbf{n}$ in (1.1) can be shown to be twice the mean curvature H of the interface. The mean curvature of a surface is the arithmetic average of the two principal curvatures κ_i ($i = 1, 2$):

$$H = \frac{1}{2} (\kappa_1 + \kappa_2) = \frac{1}{2} \left(\frac{1}{R_1} + \frac{1}{R_2} \right).$$

Each principal curvature is the curvature of a curve, i.e. the inverse of the radius of the osculating circle, defined by a plane normal to the surface. For every normal plane a different curvature can be obtained, the principal curvatures are the maximum and minimum of those values. There exists an intricate relationship between minimization of surface area and curvature as illustrated by soap films attached to a wire frame, whose equilibrium shape is described minimal surfaces, which are surfaces of zero mean curvature. It is common in fluid mechanics to define the curvature κ as $\kappa = 2H$. A theorem of differential geometry states that

the curvature of a surface Γ embedded in \mathbb{R}^3 can be computed as [39, 58]

$$\kappa \mathbf{n} = (\nabla_S \cdot \nabla_S) id_\Gamma = \nabla_S^2 id_\Gamma, \quad (1.3)$$

where id_Γ is the identity mapping on Γ . In two dimensional Cartesian coordinates (x, y) we have

$$\nabla_S id_\Gamma = \mathbf{t} \mathbf{t} \nabla id_\Gamma = \mathbf{t} \partial_s id_\Gamma = \mathbf{t} \mathbf{t},$$

where $\mathbf{t} = (t_1, t_2)^T$ is a unit tangent vector and $\partial_s = t_1 \partial_x + t_2 \partial_y$.

1.1.1 Contact Lines

If the interface bounding two fluids is not a closed surface but intersects with another interface or a solid surface, the region where the two surfaces intersect is called a contact line. The contact line will in general be a one-dimensional curve. This case is not only topologically very different from the case where the interface is a closed shape, but also physically as it gives rise to many new interesting and complicated phenomena, see [33] for a review. As fluid molecules on an interface experience different neighboring forces as those in the bulk giving rise to surface tension, molecules close to a contact line will experience even more different forces since they are now close to molecules of the three different materials.

The macroscopic system formed by a drop of liquid resting on a solid surface and surrounded by a gas or by vacuum usually forms a characteristic angle at the contact line. In this case, the angle measured through the liquid between the interface and the solid surface is called the contact angle θ , see Fig. 1.2. The contact angle quantifies the wettability of the solid surface by the liquid, where $\theta > \pi/2$ characterizes a hydrophobic surface and $\theta < \pi/2$ a hydrophilic surface. Wettability is an influential parameter in boiling heat transfer [13]. For a static

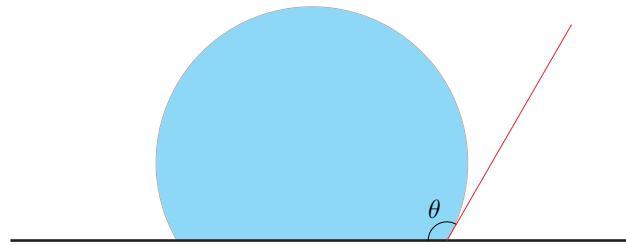


Figure 1.2: Contact angle θ formed by a drop of liquid on a plane surface.

drop, the contact angle is determined by the properties of the three materials meeting at the contact line. The equilibrium of forces at the contact line for a fluid/liquid/solid system, where

$\sigma_{gs}, \sigma_{ls}, \sigma_{lg}$ are the gas-solid, liquid-solid and liquid-gas interfacial energies respectively, leads to Young's celebrated equation:

$$\sigma_{gs} = \sigma_{ls} + \sigma_{lg} \cos(\theta). \quad (1.4)$$

If the the contact line is in motion relative to the solid, the contact angle is called a dynamic contact angle and will in general deviate from the static contact angle. The dynamic contact angle shows a highly non-trivial behavior and is in general dependent on the properties of the fluids, the solid and also on the flow field in the vicinity of the contact line [123]. Hysteresis of the dynamic contact angle is in general observed, i.e. its value will vary between the advancing and the receding contact angle values depending on whether the contact line was previously in motion. Despite being a frequently observed phenomenon that is important to many practical applications (e.g. coating processes), the physics of moving contact lines are still very poorly understood [124].

In fact, classical hydrodynamics fails to describe moving contact lines. Scriven [66] has demonstrated that it results in a non-integrable stress singularity requiring an infinite force to move the solid relative to the fluid. The problem is due to the fact that the solution of the Navier-Stokes equations has to satisfy both the kinematical condition and the no-slip condition at the contact line. While the flow kinematics include contact line motion, the no-slip condition does not allow it to move, implying that the contact line should remain pinned on the surface at all times. The later contradicts everyday experience as we have all seen rain drops sliding on a window. The non-integrable singularity can be circumvented by relaxing the no-slip condition [41]. An alternative boundary condition is the Navier slip boundary condition [93], which allows for a non-zero velocity at the contact line being proportional to the local shear-stress. Experimental visualizations have shown that the fluid motion near the contact line is described by a sort of rolling motion, i.e. fluid particles at the interface move towards the contact point and then away from it on the solid surface [123].

1.2 Finite Element Method

Two-phase flow problems are described by partial differential equations (PDE). Many numerical methods exist for the discretization of PDE. Three classical mesh based methods are finite difference methods, finite volume methods and finite element methods. Finite differences (FD) benefit from their intuitive derivation from a Taylor series expansion and their straight-forward implementation for simple problem geometries but they are difficult to extend to more complex geometries. The finite volume method (FVM) applies conservation principles for mass, momentum and energy to discrete control volumina allowing the exact integral conservation principles to be verified around every mesh cell. In the FVM, unstructured meshes can be used, which makes it flexible for handling complicated geometries and its relative ease of implementation makes it very popular for fluid dynamics computations in engineering.

An important property of discretization methods is their order of convergence, which measures how fast the discretization error decreases when the mesh resolution is increased. First and second order accurate FDM and FVM are common but their extension to higher orders is cumbersome to implement. For these methods a higher order approximation usually requires larger interpolation stencils. This not only increases the band-size of the matrix but also makes it difficult to handle boundary conditions. In the FEM on the other hand, higher order can be achieved locally inside a compact element. Another advantage of FEM is the existence of a strong theoretical background including rigorous error estimates. Moreover, FEM is well suited for unstructured meshes, which is an important requirement to discretize complex geometries. This makes the FEM a flexible and powerful tool for solving PDE in many area of physics and in particular in fluid dynamics. Unstructured meshes and the FEM are thus chosen in this work to handle the complicated interface motion arising in two-phase flow.

1.3 Motivation and Structure of the Thesis

Recent developments in the electronics industry have led to increases in heat dissipation rate. Applications such as computer data centers, electric vehicle power electronics, avionics, radars, and lasers produce amounts of heat that challenge traditional air cooling techniques [91]. Moreover, the sustained trend of miniaturization of microprocessors as described by Moore's law¹ cannot continue for ever. By the 2020s, further downscaling of current silicon based technology will lead to circuit features that are only a few nanometre across and quantum mechanical effects will make transistors unreliable [141]. One way to continue increasing computational power per unit volume is to stack many circuits on another. The success of this technology will depend on whether the heat generated by such a 3D stack can be removed efficiently. This leads to an urgent need for innovative cooling techniques to maintain chip temperatures below limits set by both materials and device reliability.

This trend is responsible for a recent transition to two-phase cooling, which exploits the latent heat of vaporization rather than sensible heat alone. Two-phase flow cooling allows to achieve high heat transfer coefficients and high heat fluxes at almost constant temperature. The latter is important to limit the thermo-mechanical stresses and can improve the durability of the equipment. Flow boiling enables very high heat transfer rates at small liquid flow rates [76], thus allowing very small liquid pumps to be used, resulting in very compact cooling systems. Three promising two-phase cooling configurations have appeared [91] as best contenders for the most demanding applications: jet impingement, spray cooling and mini/microchannels. Jet impingement and spray cooling have the potential to produce very high heat transfer coefficients in a concentrated region. Microchannel heat sinks are ideally suited for applications demanding the dissipation of large amounts of heat in very limited space [91].

The design and performance assessment of cooling systems are highly dependent on empirical predictive tools and, to a far lesser extent, mechanistic models [91]. Empirical correlations

¹Moore's law predicts that the number of transistors on a chip will double approximatively every two years.

are usually derived from large databases for different coolants, geometrical parameters, and operating conditions. The term microchannel is used to refer to channels where two-phase flow correlations change significantly due to microscale effects. The differentiation between macro and microchannels can be based on the volume to surface area ratio and channels with hydraulic diameter smaller than a few mm are usually considered as micro. The macro to micro scale transition has also been defined in terms of non-dimensional numbers where the Eotvos number and the confinement number have been used [76]. Many two-phase flow and heat transfer design methods exist for macrochannels but due to the small sizes (microchannels can be $100\mu m$ in diameter), which make it difficult to perform non-intrusive measurements, prediction methods for microchannels are fewer [111]. While for large channel sizes turbulence and droplet entrainment are important, in microchannels the flow is typically laminar and governed by viscosity, surface tension and boiling effects. The effects of viscous dissipation, which are commonly neglected in large channels, are found to increase with decreasing channel sizes [80, 90]. Viscous dissipation has the effect of increasing the temperature of the flowing fluid along the channel axis.

The most important flow patterns for thermal microchannel applications are slug flow, which is composed of long gas bubbles flowing through the denser phase and annular flow, in which the liquid is flowing in a layer on the wall forming an annulus around the gas region. When boiling occurs in a microchannel, a nucleated bubble typically grows very fast and fills the entire tube cross section [137], thus leading to an elongated slug as shown in Fig. 1.3a. When

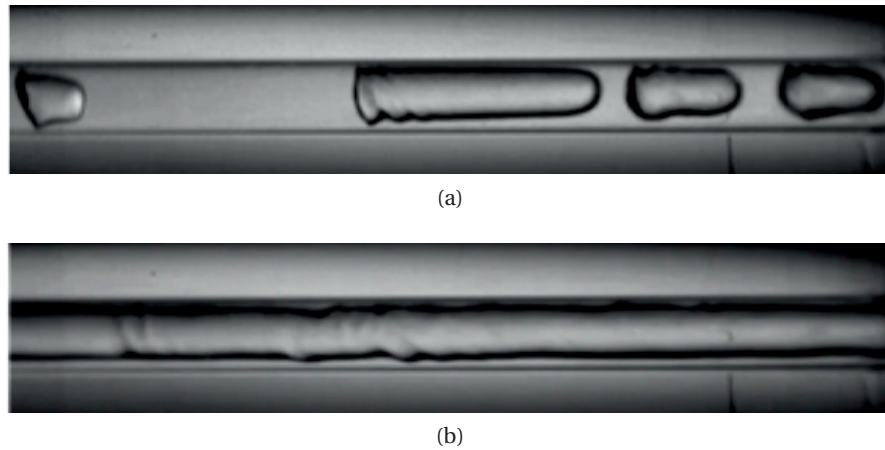


Figure 1.3: Typical flow patterns in microchannels: (a) slug flow, (b) annular flow [94].

several slugs merge, the flow transitions to the annular flow pattern displayed in Fig. 1.3b. In annular flow the interface is typically traversed by waves. These waves increase the risk of the annular liquid film drying out with very bad consequences for the heat transfer characteristics and even possible failure of the device.

Numerical simulations are an attractive tool to investigate the flow dynamics with heat and mass transfer in microchannels as they provide detailed spatial and temporal resolution, being able to describe the complete distribution of the physical quantities of interest. According to

Knudsen's criterion, the characteristic length scales of microchannels are usually sufficiently large, compared to molecular length scales, allowing a continuum description. Moreover, flow speeds are usually sufficiently low for compressibility effects to be negligible and the flow to be laminar. Two-phase flow in microchannels can thus be modeled based on the incompressible Navier-Stokes equations. In the continuum mechanics picture, the interface is infinitesimally thin and fluid properties as well as some flow quantities (like the pressure) change discontinuously over the interface. The modeling of interface dynamics and the accurate representation of the surface tension force have proven to be key issues in the numerical simulation of two-phase flow.

Based on the aforementioned background, the aim of this thesis was to develop a numerical framework for the direct numerical simulation (DNS) of two-phase flow with phase change in the capillary flow regime. Therefore, an existing moving mesh flow solver was further developed and applied to the simulation of various benchmark two-phase flow problems. The computational solver is written in the C++ programming language and solves the unsteady Navier-Stokes equations for two incompressible phases with the Finite Element Method (FEM). The contributions of the present work are

- the solver was extended to the simulation of axisymmetric flows based on the governing equations in cylindrical coordinates.
- a static contact angle and a phase change model have been included into the code allowing to simulate wetting and boiling.
- the enhanced solver was extensively validated with benchmark cases important for two-phase flow simulations.

The thesis is organized as follows, chapter 2 introduces the background on numerical modelling of two-phase flow and reviews the state of the art. Chapter 3 presents the governing equations, which are the basis for the numerical discretization presented in the following chapter. In chapter 4 the numerical procedure underlying the current solver approach is presented in a detailed way. Chapter 5 is devoted to numerical applications of the solver, where numerical results for a variety of test case are presented and compared to analytical solutions or experimental data. Conclusions and further recommendations are given in chapter 6.

2 Numerical Modelling of Two-Phase Flows: An Overview

Over the last decades, the use of numerical methods to approximate the solution of physical problems, in particular in the context of fluid flow simulation, has grown into a major research domain. Numerical methods have been applied to simulate complex physical phenomena like drop splashing, atomization processes and ocean waves [47]. This research field is expected to play a key role in the future due to the large cost saving potential relative to experiments. However, many problems of highest interest still require too much demand on computational resources, even for today's computers. This is the case for problems involving a wide range of spatial and time scales, like turbulent flows. Moreover, numerical methods always bear the danger of producing non-physical results. Therefore, careful comparison with experimental or analytical data is always necessary to validate any new numerical simulation tool.

For most problems in engineering and nature, the macroscopic view based on the integral balances of mass, momentum and energy has been very successful. However, there are situations where molecular effects become important, thus requiring a microscopic or molecular view. Strictly speaking, the continuum approach is valid only if the mean distance a molecule travels between two interactions with other particles is very small compared to any characteristic length of the flow problem. Computational methods can be distinguished based on how much of the microscopic (molecular or atomistic) world is modeled. On the one extreme are molecular dynamics (MD) simulations, which follow the molecules individually. MD simulations require way too many degrees of freedom for typical engineering problems but they are very useful in material science. In this chapter we review some of the computational methods that are commonly applied for engineering simulations. We start with methods that include some microscopic effects (meso-scopic view) and proceed with the continuum mechanics models, which are used in this work.

2.1 Meso-Scopic Models

Diffuse-interface models offer a way to account for the additional physics that become relevant when the thickness of the interfacial layer becomes comparable to the length scale of

the phenomena being examined, see [1] for a review. They are based on statistical mechanics, kinetic theory and non-equilibrium thermodynamics. Diffuse-interface models replace the sharp interface by a thin but finite transition region and quantities that are local in the continuum formulation (like surface tension) become distributed throughout the interfacial region. Such models have been used to simulate moving contact lines and phase change phenomena. Examples of situations involving physical mechanisms acting at scales comparable to the interface thickness are: near the critical point where the interface thickness diverges, the motion of the contact line, breakup of fluid threads and merging of bubbles. A finite width of the interfacial layer is inherent to diffuse-interface models. By considering the asymptotic limit in which the ratio of interfacial width to macroscopic length scale vanishes (i.e. the sharp-interface limit), the diffuse-interface models can be related to the continuum equations of fluid dynamics [1].

An example of diffuse-interface models are phase Field methods, which are based on an order parameter (or phase-field variable) and an equation for the free energy as a functional of the phase-field variable [1]. The phase-field variable is related to the volume fraction of the individual components and its evolution is governed by the Cahn-Hilliard equation, which is a fourth order differential equation. The equilibrium interface profile can be found by minimizing the free energy functional with respect to variations of the phase-field variable [7]. The evolution of the phase-field variable is coupled to the Navier-Stokes equations for the macroscopic variables (usually velocity, pressure and temperature). Phase field methods usually require a very thin interfacial layer to model the physics of the problem, leading to high computational requirements to resolve the sharp gradient in the transition region [7]. Moreover, due to the high order differential operators involved, phase-field computations typically use high order discretization methods, such as: spectral methods [7] or isogeometric analysis (IGA) [54, 85]. The phase-field method was used in [73] to simulate an evaporating bubble in a microchannel, investigating the effects of the contact angle and the results were compared to experimental data. In [85] the Navier-Stokes-Korteweg equations were solved using IGA to simulate bubble coalescence in annular geometry and a liquid droplet spreading on a surface.

The lattice Boltzmann method (LBM) is a computational method that solves the Boltzmann equation of statistical mechanics. The underlying idea of LBM is to use simplified kinetic models for the microscopic or meso-scopic processes such that the averaged properties obey the desired macroscopic equations [25]. The LBM evolved from lattice gas automata (LGA), which use a lattice discretizing both the position and velocity space. Every point of the lattice is connected to its neighbors by a finite number of velocity-links. In LGA there can be either zero or one particle at every lattice point occupying a particular velocity-link. An LGA computation is composed of two steps, first the particles are propagated with their corresponding velocity and then collision rules are applied. Suitable collision rules should conserve the particle's mass, momentum and energy. Momentum and energy are thus transferred in discrete steps from fluid particle to fluid particle, rather than in a continuous way. Lattice gas automata are able to approximate fluid motion if an ensemble average is taken but they

suffer from statistical noise among other issues. To get rid of the statistical noise of LGA, the LBM replaces the Boolean particle number by a probability distribution. The LBM uses a lattice to discretize the phase space (position and velocity space) and it applies a sequence of collision and streaming steps. The dependent variable of LBM is the probability density function, which has to satisfy the Boltzmann equation with a simplified collision operator describing a relaxation towards an equilibrium distribution. The macroscopic variables can be obtained from the probability density function and used to compute the equilibrium Maxwell-Boltzmann distribution, which in turn enters the collision term. It can be demonstrated, via the Chapman-Enskog multiscale expansion, that the lattice Boltzmann equation reduces to the macroscopic Navier-Stokes equations for vanishing lattice spacing. Models for the interactions between different fluid phases can be included in the LBM [121] to simulate multiphase flow, with phase separation resulting from the particle dynamics without any special treatment being needed for the interfaces. The LBM was used for the simulation of capillary waves [59] showing good agreement with the dispersion relation from linear theory. Since LBM can easily incorporate solid boundaries of complex geometry [59], Lattice Boltzmann multiphase fluid models have been used to simulate multicomponent flow through porous media. Phase change, wall wettability and roughness effects can also be included in LBM. Blake et Al [16] simulated dynamic wetting with the LBM. In [55] an interparticle interaction force scheme was proposed for LBM and used to simulate droplet motion and coalescence on bottom surfaces with wettability gradients. The LBM was used in [37] to investigate the bubble nucleation and departure within a microchannel.

2.2 Macroscopic Models

In this section we review numerical approaches based on the continuum equations of fluid mechanics. Several computational methods have emerged to solve the Navier-Stokes equations with multiple phases and no particular method has proven superior in all situations. Here, we focus on direct numerical simulation (DNS), where the flow field is computed by solving the governing equations while directly resolving all the interesting temporal and spatial scales. The beginning of multiphase flow simulation can be traced back to the second half of the 20th century when Harlow et Al. [61] presented a finite difference discretization of the Navier-Stokes equations, with a fixed computational mesh, advecting marker particles to keep track of the free surface. In the continuum limit, two-phase flow is characterized by the presence of an interface, which mathematically is a surface in three space dimensions or a curve in two space dimensions. Numerical modelling of two-phase flow can be difficult due to complex evolutions of interface and changes of its topology. Among the most recognised numerical issues are: the robust representation of the evolving interfaces, the accurate computation of the surface tension force and handling large density/viscosity ratios [47]. The second point is of particular importance as some numerical methods give rise to non-physical spurious currents [110], which sometimes may even prevent convergence of the solution. Spurious currents, generated by finite elements discretizations, have been studied thoroughly

for interface resolving [27] and non-resolving meshes [48].

Two approaches exist to include the surface tension force into the fluid mechanics equations: the "two fluids" approach solves the equations for each phase separately and surface tension modifies the boundary condition at the interface between the phases, while the "one fluid" approach introduces surface tension as a local source term in the momentum equation. For the latter the continuum surface force (CSF) method [18] has been widely used. Both approaches have been compared for the simulation of a static drop in equilibrium in [148]. It was concluded that the "two fluids" approach is algorithmically more demanding but more accurate while the "one fluid" approach offers a greater flexibility but is particularly prone to spurious currents. In the present work the "one fluid" approach is used but we will show that, as discussed in [99, 27], spurious currents can be eliminated by an appropriate discretization of pressure gradient and surface tension term, provided the curvature is computed exactly.

Numerical methods can be classified into mesh-based and mesh-less methods. Mesh-less methods are usually based on particles, like the smoothed Particle Hydrodynamics (SPH) method [17]. Particle methods are known to suffer from lack of accuracy to represent the shape of the interface but they are very flexible in handling complex fluid phenomena such as splashes, breaking waves and topological changes. Particle methods are widely used for astrophysical simulations [89] and graphical animations due to their efficiency in quickly simulating complex fluid-like behaviour. They even allow simulations to be carried out in real time [147], which is a key requirement for certain applications such as computer games.

Mesh-based methods can be further classified into interface capturing and interface tracking. Interface capturing methods make use of the Eulerian description and use a fixed mesh on which they solve an equation for the advection of a "color" function describing the different fluids and the interfaces. Mainly two types of color function are used: the volume fraction, leading to the Volume Of Fluid method (VOF) [53, 63, 110, 99], or a signed distance function, in the Level-Set method [130, 58]. In the VOF method the color function represents the volume of one phase present in each computational cell. The VOF method is able to conserve the respective mass of the phases exactly. However, its representation of the interface suffers from low accuracy, which has bad repercussions on the computed curvature and the surface tension force. The Level-Set method instead, is accurate to compute the curvature, but it does not conserve mass very well. Moreover, a frequent reinitialisation is required to keep the Level-Set a signed distance function [58]. To take advantage of the respective advantages of the VOF and Level-Set methods, some researchers have proposed to use a combination of both [11].

2.2.1 Interface Tracking

Interface tracking methods describe the interface by a set of particles or interconnected mesh points, which are either part of a secondary mesh advected relative to a fixed primary mesh (background mesh) or are a subset of the primary mesh. In the former approach the flow equations are solved on an Eulerian background mesh and the interface velocity is

interpolated from the background mesh. Examples for this approach are the front-tracking (FT) method [139, 75] and the immersed boundary method (IBM) [125], which track three-dimensional interfaces using triangulated surfaces and interpolates the interface velocity from the background mesh. In [125], an algorithm was presented for maintaining the interface connectivity information when topology changes occur and used to simulate a rising bubble and bubble/drop coalescence. Another example is the sharp-interface method (SIM) [148], where each phase has its own governing equations connected via boundary conditions at a discontinuous interface.

The Lagrangian approach consists in moving the mesh nodes with the local flow velocity. However, fully Lagrangian approaches often result in severe mesh distortions that can result in mesh tangling making them impracticable for most problems unless continuous remeshing is used. A Lagrangian finite element method has been used in [77], with a mesh rezoning technique, to compute the effects of varying water depth on the propagation of a solitary wave. In [109] a similar problem was investigated where a water wave was shoaling over a constant slope topography with eventual wave breaking. Arbitrary Lagrangian-Eulerian (ALE) methods allow to avoid the shortcomings of purely Lagrangian approaches by moving the mesh in an arbitrarily specified way [35, 36]. In general, the Lagrangian approach (mesh moving with the flow velocity) is required to move the interface mesh. The ALE approach allows for a movement of the interface mesh in the Lagrangian way while the remaining mesh nodes may be adjusted in an arbitrary manner, with the goal of preserving well shaped mesh elements during the simulation. Moving mesh methods are useful for problems displaying moving boundaries/layers, scale invariance or self similarity [8]. The ALE method is well suited for moving boundary problems in general and in particular for simulations of fluid-structure interactions [146].

Interface capturing methods have the advantage of being flexible with regard to complicated interface motion because topological changes are handled automatically, although not in a physical way. However, in fixed mesh methods, interfaces are in general not resolved by the mesh thereby reducing the accuracy of interfacial tension and the representation of the discontinuous material properties. Interface tracking methods are able to resolve the interface with edges/faces and nodes of the computational mesh, thus making the interface a true discontinuity. Therefore, interface tracking allows for a more accurate representation of the interface and the curvature calculation, but this comes at the price of a reduced flexibility to handle complicated interface deformation. Since the interface is tracked explicitly, topological changes such as breaking and merging of interfaces have to be "included manually" when using interface tracking or front tracking methods. Moreover, guarantying mass conservation can be an issue for front tracking methods where the interface is advected by a velocity, which is interpolated from the surrounding mesh nodes and is, in general, not divergence free. This mass conservation problem is absent from ALE methods where the velocity is defined at the interface nodes, thus avoiding the need for interpolation. Table 2.1 summarizes the main advantages and drawbacks of interface tracking and interface capturing approaches.

Chapter 2. Numerical Modelling of Two-Phase Flows: An Overview

Interface Tracking	Interface Capturing
+accurate representation of the interface -topological changes need to be "included manually"	+topological changes happen automatically -additional equation for color function

Table 2.1: Comparison of interface tracking and interface capturing approaches.

The finite element method (FEM) has shown to be well suited for moving mesh methods for the same reasons which recommend it for fixed-mesh applications [86]: flexibility for irregular geometry arising from nodes motion is easily accommodated and representation of variable coefficients is possible. Moreover, the FEM allows for a particular kind of interface tracking. Space-time methods [14, 88, 135] discretize both the space and time domain simultaneously with the FEM. As a result, the free boundary or interface motion is automatically accounted for by the shape of the space-time elements. Another interesting approach is the eXtended Finite Element Method (XFEM), which uses a fixed mesh with a shape function space that is locally enriched by functions that enable the exact approximation of discontinuities. By adding discontinuous interpolation functions to the elements traversed by the interface, the XFEM method is able to represent a discontinuous interface on a fixed mesh. In [116], different enrichment schemes are investigated for the velocity and pressure spaces, in the context of two-phase and free-surface flow simulations. However, a drawback of the XFEM approach is that it suffers from ill conditioning of the resulting linear system, adding stress on the linear equation solver [115].

In the classic ALE method the time derivative is expressed with respect to a deforming configuration and the mesh velocities enter the flow equations via the convective term [62, 36, 69]. Besides the dependent variables of the flow problem, deforming mesh methods introduce many degrees of freedom associated with the trajectories of the mesh nodes. Mesh nodes at the interface have to satisfy the kinematic conditions and hence their velocity is specified while the mesh points on the boundaries of the domain can usually remain fixed. However, the motion of mesh points which are "in between" (the volume mesh points) remains to be specified, with the objective to preserve a good mesh quality. Many ALE schemes use mesh velocities generated by mechanical analogies. A popular approach is to use a mesh which deforms like a fictitious elastic body [26, 134, 74, 49, 86] subjected to displacement boundary conditions at the interface and domain boundary. Several mathematically motivated approaches also exist [8], which try to optimise geometric qualities of the mesh by minimizing a functional or find the mesh velocities as a solution to an additional differential equation.

An important question, in the framework of moving mesh schemes, is the role of the Geometric Conservation Law (GCL), which is discussed in [136]. The GCL originated from the finite volume framework, where it provides a condition to maintain the inherent conservation properties of the FVM. Starting from the integral GCL, Thomas and Lombard [136] derive a differential version of the GCL, which has to be satisfied by a finite difference scheme based

on a coordinate mapping from a physical moving boundary domain to a time independent reference domain. The authors argue that the GCL condition has to be satisfied in order to avoid additional errors in the flow variables. The GCL is usually derived from the requirement [136] that starting from a uniform flow, the numerical scheme must be able to reproduce this uniform flow exactly at all times on a moving mesh. Cao et Al [23] used the GCL to derive a mesh movement algorithm. They impose the Jacobian of the moving domain transformation based on a monitor function and derive a differential equation for the mesh velocity field, which is then shown to be equivalent to the minimization of a functional. A similar method was proposed by Baines et Al [10] based on a local conservation principle to find the mesh velocities. In the so called conservation method, the nodes of the mesh are moved such that the mass fraction associated with particular subregions is preserved. A detailed description of the conservation method is given in [8], along with its application to diverse moving boundary problems. The conservation method has been used to simulate glacier movement [95] and tumour growth in [83], where it was shown to have higher order of convergence than transforming the problem to a fixed domain. In [9] single and two-phase Stefan problems were simulated using only the energy equation (without coupling to the Navier-Stokes equations) and it was shown to converge towards the analytic solution with second order accuracy. In the aforementioned papers, the conservation method was used without changing the number of mesh nodes or their connectivity. Therefore, the mesh tangled when the domain occupied by one phase became too short.

Severe mesh deformations can occur with interface tracking methods and in the worst case, some mesh elements might even be deformed to a point that prevents the simulation to proceed. Therefore, remeshing can help to handle cases where the motion of the interface would otherwise lead to strong mesh distortions. Remeshing can mean changing only the connectivity of the mesh and/or inserting/removing mesh points. After remeshing, an interpolation is required to project the solution from the old onto the new mesh. This interpolation introduces additional errors, which are usually of diffusive nature, i.e. they tend to smear out steep gradients. Some authors propose changing only the local connectivity, e.g. by swapping edges and avoid global remeshing [31, 106]. However, at the implementation level, these techniques lead to frequent insertion/deletion operations on vectors, which typically require doubly or single linked lists to perform better and may become less efficient than global remeshing. Moreover, remeshing offers the opportunity to adaptively refine the mesh, increasing the mesh resolution where needed while keeping a coarse mesh elsewhere. In [92], it was demonstrated with a simple example that remeshing at every time step can be more accurate than using an ALE scheme without remeshing.

Free surface flow was simulated in [19] using the FEM and the ALE approach with a node repositioning algorithm based on the minimization of a penalty function. There, surface tension was modelled by minimizing the surface energy and it was argued that the ALE approach could compete with fixed mesh approaches such as the VOF. An ALE method with pseudo elastic mesh update and remeshing to handle large deformations is described in [142] for three dimensional free surface flow problems including dynamic contact lines. The

contact line speed is computed as a function of the contact angle, the capillary number and the static contact angle to simulate gravity driven motion of a droplet along an inclined plane and the formation of a meniscus on a tetrahedral mesh of isoparametric quadratic finite elements. In [106] an interface tracking method with an adaptive unstructured mesh has been proposed for three-dimensional two-phase flow including a scheme to simulate the breakup of a fluid thread. The curvature was computed by a least squares parabola fitting and the method was used to simulate several test cases: two-phase Couette flow, oscillating droplet, a droplet in shear flow, a pinching ligament and modulated jet pinching. The method was then extended [107] to interface merging/coalescence and used to simulate droplet collisions. An ALE moving mesh method discretizing the axisymmetric flow equations by the FEM was presented in [51, 49] to simulate free surface and two-phase flow problems including static bubble, oscillating droplet, liquid droplet impinging on a solid surface and a rising bubble. The curvature was approximated using the Laplace-Beltrami operator technique combined with a boundary approximation using isoparametric finite elements. This method was shown to produce almost no spurious currents [48] and to handle problems with large jumps in material parameters without loss of accuracy. Moreover, the Laplace-Beltrami technique allows for a weak enforcement of the equilibrium contact angle as shown in [50] where the spreading of a pendant liquid droplet was simulated with a Navier-slip boundary condition at the contact line.

The computational code proposed in [3], which is the starting point for the present work, implements a combination of the numerical techniques described above. It uses the FEM to discretize the two-phase domain with an interface resolved moving unstructured mesh within the ALE framework like in [51, 49] and mesh maintenance algorithms that are largely inspired by [106] including Laplacian smoothing as proposed in [31]. Compared to choosing mesh that deforms like a fictitious elastic body, the mesh motion scheme of [3] has the advantage that no additional elasticity problem needs to be solved. The method was extended to the cylindrical axisymmetric flow equations in [56] and validated versus analytical and experimental results. The interface motion is tracked by moving the mesh nodes and remeshing is used making it able to handle cases with large mesh deformation but without topological changes. The time required by the remeshing procedure and the successive interpolation was found to account only for a small fraction in the time of the overall algorithm. The present method is accurate in its representation of the curved interface and in order to avoid spurious currents the same FEM basis functions are used for the pressure and the surface tension term [27], as will be discussed in Section 5.4.

2.2.2 Phase Change

In this section we briefly review the state of the art in numerical methods for two-phase flow including phase transition. The reader is referred to [78] for a more comprehensive review. Phase change introduces further issues for numerical algorithms like the normal velocity discontinuity, the accurate evaluation of the phase change mass flow rate and sharp

boundary layers of the temperature field close to the interface. Early moving mesh finite element methods have been applied to the Stefan problem, solving only for the temperature and interface position. In [14] two-dimensional Stefan problems were solved with space-time finite elements and in [86] freezing around a pipe was simulated. However, it appears that the first numerical simulation of the Navier-Stokes coupled to the energy equation with phase change was pioneered by the work of Welch [143]. Welch used a moving unstructured mesh with double degrees of freedom at the interface to simulate evaporation in a closed environment, thermocapillary motion, an oscillating bubble and nucleate boiling. In a second paper [144] Welch, simulated the growth of a vapor bubble on a surface with a pinned contact line. In all these early works, no remeshing was used thus limiting their range applicability to cases without strong interface deformations.

As for the surface tension force, the phase change mass transfer can be included in the interface jump condition or as a singular source term acting at the interface. The SIM method with cut cells (SIMCC) is based on the former approach and it has been used in [132], with a quadratic curve fitting algorithm for the curvature, to simulate the cryogenic flow of a bubble in a tube with phase change. Most computational codes for phase change use the source term approach, with many authors using the VOF method [60, 118, 145]. Son and Dhir [126] modified the level-set method to include liquid-vapor phase change and investigated film boiling. A coupled level-set and VOF method has been used by Tomar et Al [138] to simulate film boiling and investigate the frequency of bubble formation. The Front Tracking (FT) method has been adapted to phase change problems and applied to simulate phase change by Tryggvason and coworkers [75, 44, 140]. The FT method has been used to simulate the Stefan problem [43, 44], 3D film boiling of a liquid sheet [44], boiling around a cylinder [140], boiling around multiple cylinders [45] and explosive boiling in microgravity [43].

Some methods may be less suitable to handle the discontinuous normal velocity at the interface, which are caused by the phase change mass source term. In the FT method, interpolation errors might make it difficult to conserve the mass of the respective phases. Among some recent developments in numerical methods for phase change are the mass-conservative Cartesian grid method, developed by Sato et al [114], which was used to simulate an evaporating rising bubble and nucleate boiling. The XFEM method was used in [87] for solidification problems and the results were compared with ALE simulations using the commercial software COMSOL multiphysics. COMSOL's ALE moving mesh method was also used in [72] for the simulation of evaporating bubbles in a microchannel but with very little previous validation.

A fundamental question in phase change computations is how to model the mass transfer rate at the interface. Two main categories of models can be distinguished: those based on thermodynamic equilibrium, where the mass transfer rate is computed from the energy jump condition [44, 114, 126, 132, 143, 144, 145] and models accounting for a departure from thermodynamic equilibrium [60, 72] where the mass transfer is computed by a formula derived from molecular gas kinetic considerations. Both models respect the energy jump condition, which follows from the integral energy conservation principle. However, the first kind of

models assume that the interface is at saturation conditions based on local equilibrium such that the temperatures in both phases are equal at the interface. Models of the second kind allow the temperature field to be discontinuous at the interface and compute the phase change mass transfer, based on kinetic theory, following the work of Schrage [119] and Tanasawa [133].

In this work, we assume the interface temperature is continuous and equal to the saturation temperature. The latter is supposed uniform as we consider the mono-component case and constant based on the isobaric approximation. This is a simplification over physical reality where a small interface superheat and interface thermal resistance are usually observed. More specifically, at a curved interface where the liquid and vapor pressures cannot be equal due to the capillary pressure jump, there will be deviation from equilibrium. In [75] the choice of the interface condition for the temperature is discussed for the simulation of film boiling and compared to experimental data. While non-equilibrium thermodynamics models are aimed at a better description of the actual physics of phase change, such models usually contain phenomenological constants, like the accommodation coefficient, whose quoted values in the literature vary wildly [24] because it is hard to measure experimentally [104]. As the exact value of these constants is unknown even for common fluids like water they represent free parameters, which are assigned an a priori arbitrary value in computational studies [78]. Since it allows to model phase change in a self consistent way, remaining inside the framework of continuum mechanics, the thermodynamic equilibrium approach has been chosen in the present work. Therefore, we solve the energy equation by setting the temperature of the interface equal to the saturation temperature and use the energy jump condition to ensure that mass and energy transfer at the interface due to phase change are correctly accounted for.

3 Governing Equations

This chapter introduces the equations describing the motion of two in-miscible, incompressible Newtonian fluids with phase change. The equations presented in this section have been solved numerically with the methods described in the next chapter. In this chapter we will first present the governing equations for mass, momentum and energy without phase change. Phase change will then be introduced in the last section.

In continuum mechanics a two-phase flow is considered as a field, which is divided into regions (the phases), and the interface separating the different phases is a moving boundary. In general, the governing equations of single phase flow, expressing the balance of mass, momentum and energy, hold for each region. Additionally, appropriate matching conditions need to be prescribed at the interface. One can distinguish between interface jump conditions and interface boundary conditions [71], where the first specify exchange of mass, momentum and energy through the interface. The interface jump conditions can be derived from the general form of the conservation equations of the bulk fluid phases applied to a control volume that includes the interface. However, for viscous flow, they are insufficient to have a well posed problem and they need to be supplemented by interface boundary conditions, which are of a more phenomenological nature [104]. Examples of the latter are the conditions of continuity of the tangential component of fluid velocity and the temperature at the interface.

3.1 Mass and Momentum Equations

In this thesis, we solve the Navier-Stokes equations in velocity-pressure formulation. Two equivalent formulations of the non-dimensional Navier-Stokes equations are going to be presented next. The dependent flow variables are the fluid velocity $\mathbf{v}(\mathbf{x}, t)$ and pressure $p(\mathbf{x}, t)$, which are functions of spatial coordinate \mathbf{x} and time t . In the following, lengths have been non-dimensionalized by a characteristic length L , velocities by a characteristic velocity V and time by a characteristic time scale L/V . The non-dimensional quantities are related to the

Chapter 3. Governing Equations

corresponding dimensional quantities (with subscript dim) by

$$\mathbf{v} = \frac{\mathbf{v}_{dim}}{V}, \quad x = \frac{x_{dim}}{L}, \quad t = \frac{t_{dim}V}{L}, \quad (3.1)$$

$$p = \frac{p_{dim}}{\rho_l U^2}, \quad \rho = \frac{\rho_{dim}}{\rho_l}, \quad \mu = \frac{\mu_{dim}}{\mu_l}, \quad (3.2)$$

where the subscript l denotes the quantities of the liquid phase. By virtue of this non-dimensional approach two dimensionless numbers are introduced. They are the **Reynolds number** defined as:

$$Re = \frac{\rho_l V L}{\mu_l}, \quad (3.3)$$

which expresses the ratio of inertial effects to viscous effects and the **Froude number**:

$$Fr = \frac{V}{\sqrt{g_0 L}}, \quad (3.4)$$

which gives the ratio of inertial effects to gravitational effects.

3.1.1 Two Fluids Formulation

In the two fluids formulation the Navier-Stokes equations are satisfied for each phase individually with coupling conditions at the interface between the phases. In this case the governing equations inside every fluid phase are identical to those for a single phase. The momentum equation, expressing Newton's second law of motion, reads:

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \frac{1}{Re} \nabla \cdot [\mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T)] + \frac{\rho}{Fr^2} \mathbf{g}, \quad \text{for } \mathbf{x} \in \Omega, t \in (t_0, t_e) \quad (3.5)$$

and the continuity equation accounting for conservation of mass and incompressibility is:

$$\nabla \cdot \mathbf{v} = 0, \quad \text{for } \mathbf{x} \in \Omega, t \in (t_0, t_e) \quad (3.6)$$

where \mathbf{g} is the gravitational acceleration vector ($\mathbf{g} = \mathbf{g}_{dim}/g_0$). The material derivative of the fluid velocity:

$$\frac{D\mathbf{v}}{Dt} = \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}, \quad (3.7)$$

includes a local time derivative (first term) and a non-linear convective part (second term). The density ρ and viscosity μ are constant for each fluid. We consider two fluids occupying two separate regions of space Ω_1 and Ω_2 where $\Omega_1 \cap \Omega_2 = \emptyset$ and an instance of the equations (3.5-3.6) has to hold in each fluid region Ω_i with the local fluid properties ρ_i, μ_i .

The bulk equations (3.5-3.6) are supplemented by interface conditions. Applying the integral

principle of conservation of mass to a control volume around the interface yields an interface jump condition, which reads as

$$[\rho(\mathbf{v} - \mathbf{v}_\Gamma)]_\Gamma \cdot \mathbf{n} = 0 \quad \text{for } \mathbf{x} \in \Gamma, \quad (3.8)$$

where \mathbf{n} is a unit vector normal to the interface, \mathbf{v}_Γ is the interface velocity and $[f]_\Gamma = f_1 - f_2$ represents the jump of a generic quantity f across the interface Γ . If there is no mass transfer through the interface, Eq. (3.8) requires the normal component of the velocity to be continuous at the interface and equal to the interface velocity:

$$\mathbf{v}_1 \cdot \mathbf{n} = \mathbf{v}_2 \cdot \mathbf{n} = \mathbf{v}_\Gamma \cdot \mathbf{n} \quad \text{for } \mathbf{x} \in \Gamma. \quad (3.9)$$

It is worth noting that, despite being needed to close the system, no condition for the tangential velocities at the interface can be deduced from conservation laws [104]. Since molecular interactions are expected to be as effective in equalizing the local velocities at an interface as they are in equalizing the velocities at two neighboring points inside the bulk fluids [5] no-slip between the phases is usually assumed for viscous fluids. As a result, the fluid velocity is continuous at the interface:

$$[\mathbf{v}]_\Gamma = \mathbf{0} \quad \text{for } \mathbf{x} \in \Gamma. \quad (3.10)$$

Applying an integral momentum balance across the interface leads to a jump condition, which is called the dynamic interface condition. In the absence of mass transfer through the interface, the dynamic condition reads as

$$[-Ip + \frac{\mu}{Re}(\nabla \mathbf{v} + \nabla \mathbf{v}^T)]_\Gamma \mathbf{n} = \frac{\kappa}{We} \mathbf{n} \quad \text{for } \mathbf{x} \in \Gamma, \quad (3.11)$$

with the stress acting on both sides of the interface on the left hand side and the surface tension force on the right hand side. In Eq. (3.11), κ is the double of the mean curvature of the interface and is simply referred to as the curvature hereafter. We have introduced the **Weber number** defined as

$$We = \frac{\rho_l L V^2}{\sigma}, \quad (3.12)$$

which gives ratio of inertial effects to surface tension effects. The surface tension coefficient σ is assumed to be constant in the present context thus excluding Marangoni flows. A further non-dimensional number that will be used in this thesis is the **Eötvös number**, which is defined as

$$Eo = \frac{\rho_l g L^2}{\sigma} = \frac{We}{Fr^2}, \quad (3.13)$$

and expresses the ratio of gravitational effects to surface tension effects. The Eötvös number is useful to describe cases where no characteristic velocity exists, like for sessile droplets or

Chapter 3. Governing Equations

rising bubbles.

Normal and tangential stress conditions are obtained by multiplying Eq. (3.11) by the interface normal and tangential vector respectively. The normal stress condition is

$$\mathbf{n}[-Ip + \frac{\mu}{Re}(\nabla \mathbf{v} + \nabla \mathbf{v}^T)]_{\Gamma} \mathbf{n} = \frac{\kappa}{We} \quad (3.14)$$

and the tangential condition is

$$\mathbf{t}[-Ip + \frac{\mu}{Re}(\nabla \mathbf{v} + \nabla \mathbf{v}^T)]_{\Gamma} \mathbf{n} = 0 \quad \Rightarrow \quad \mathbf{t}[\mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T)]_{\Gamma} \mathbf{n} = 0. \quad (3.15)$$

where \mathbf{t} is a unit vector tangential to the interface. The tangential stress condition implies that the shear stress is continuous across the interface (3.15). If there is no surface tension ($We \rightarrow \infty$) and the pressure difference is zero Eq. (3.14) gives:

$$\mathbf{n}[\mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T)]_{\Gamma} \mathbf{n} = 0,$$

which implies that the normal derivatives of the velocity are discontinuous at the interface exactly when $\mu_1 \neq \mu_2$. In a static fluid, there are no viscous stresses and the normal stress condition (3.14) reduces to the celebrated Young-Laplace equation for the pressure jump across a static interface:

$$-[p]_{\Gamma} = -\Delta p = \frac{\kappa}{We} = \frac{1}{We} \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \quad (3.16)$$

where R_1, R_2 are the principal radii of curvature and the sign in Eq. (3.16) is such that for a bubble/drop, the inner region has a higher pressure. Therefore, in two-phase flow, the derivatives of the velocity and the pressure can be discontinuous at the interface.

Equations (3.5) and (3.6), hold in each Ω_i for the local fluid velocity $\mathbf{v}_i(\mathbf{x}, t)$ and pressure $p_i(\mathbf{x}, t)$, $i = 1, 2$. Together with the interface conditions (3.10, 3.11) and appropriate boundary conditions at the flow boundaries they form a closed system. The system can be solved for an initial condition $\mathbf{v}_i(\mathbf{x}, t_0)$ and an initial interface position to obtain the flow field and the position of the interface at later times: $t > t_0$. The interface position is found by integrating the interface velocity \mathbf{v}_{Γ} given by Eq. (3.9) in time.

3.1.2 One Fluid Formulation

As an alternative to the above approach, one can define a global velocity which is equal to the local velocity in each phase: $\mathbf{v} = \mathbf{v}_i$ in Ω_i , $i = 1, 2$ and replace the coupled two-fluid system by a single set of equations for both phases. Due to the interface condition (3.10) the global velocity is continuous in $\Omega_1 \cup \Omega_2$ and the stress boundary condition (3.11) can be accounted for by a singular source term \mathbf{f}_{st} . This approach is commonly termed the "one fluid" formulation,

where the equations to be satisfied are:

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \frac{1}{Re} \nabla \cdot [\mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T)] + \frac{\rho}{Fr^2} \mathbf{g} + \frac{1}{We} \mathbf{f}_{st}, \quad \nabla \cdot \mathbf{v} = 0, \quad (3.17)$$

for $\mathbf{x} \in (\Omega_1 \cup \Omega_2)$, $t \in [t_0, t_e]$ and

$$\rho(\mathbf{x}, t) = \rho_i, \quad \mu(\mathbf{x}, t) = \mu_i \quad \text{for } \mathbf{x} \in \Omega_i(t) \quad i = 1, 2. \quad (3.18)$$

The additional body force in Eq. (3.17), is expressed as

$$\mathbf{f}_{st} = \kappa \delta \mathbf{n}, \quad (3.19)$$

using the Dirac delta distribution δ to localize the interface.

The Dirac distribution can be represented as the derivative of a heaviside step function H , which is equal to zero in one phase and one in the other phase:

$$\delta \mathbf{n} = \nabla H, \quad \delta = |\nabla H|. \quad (3.20)$$

Moreover, δ has the property that it transforms a volume integral over a control volume \mathcal{V} containing the interface Γ into a surface integral over the interface:

$$\int_{\mathcal{V}} \delta dV = \int_{\Gamma} dS. \quad (3.21)$$

The present numerical method is based on the "one fluid" approach. The position of the interface and its curvature κ are coupled to the Navier-Stokes equation Eq. (3.17).

It can be shown that the "one fluid" formulation is consistent and equivalent to the two Navier-Stokes formulation. In [57] the "one fluid" formulation is derived starting from the two Navier-Stokes. Here we show the reverse namely how the two Navier-Stokes can be recovered from the "one fluid" formulation. When integrating Eq. (3.17) over an arbitrary control volume that is totally contained in either phase Ω_i , $i = 1, 2$ the surface tension term vanishes and Eq. (3.5) is obtained. Since the control volume is arbitrary this implies that Eq. (3.5) has to be satisfied everywhere in Ω_i , $i = 1, 2$. Integrating Eq. (3.17) over a control volume \mathcal{V} that includes the interface yields

$$\begin{aligned} \int_{\mathcal{V}} \rho \frac{D\mathbf{v}}{Dt} dV &= \int_{\mathcal{V}} \nabla \left(-Ip + \frac{\mu}{Re} [\nabla \mathbf{v} + \nabla \mathbf{v}^T] \right) dV + \int_{\mathcal{V}} \frac{\rho}{Fr^2} \mathbf{g} dV + \int_{\mathcal{V}} \frac{\kappa \delta}{We} \mathbf{n} dV \\ &= \int_{\partial \mathcal{V}} \left(-Ip + \frac{\mu}{Re} [\nabla \mathbf{v} + \nabla \mathbf{v}^T] \right) \mathbf{n} dS + \int_{\mathcal{V}} \frac{\rho}{Fr^2} \mathbf{g} dV + \int_{\Gamma} \frac{\kappa}{We} \mathbf{n} dS, \end{aligned} \quad (3.22)$$

where Gauss's divergence theorem has been used to transform the volume integral of the stress tensor into a surface integral and the property (3.21) has been used for the surface tension term. Letting the thickness of the volume \mathcal{V} in a direction orthogonal to the interface shrink to zero will make its boundary $\partial\mathcal{V}$ "adhere" to the interface from both sides. As a result the volume integrals in Eq. (3.22) will vanish since the velocity field and gravity are continuous functions and the following equation will result:

$$\int_{\Gamma} \left[Ip - \frac{\mu}{Re} (\nabla \mathbf{v} + \nabla \mathbf{v}^T) \right]_{\Gamma} \mathbf{n} dS + \int_{\Gamma} \frac{\kappa}{We} \mathbf{n} dS = \int_{\Gamma} \left(\left[Ip - \frac{\mu}{Re} (\nabla \mathbf{v} + \nabla \mathbf{v}^T) \right]_{\Gamma} + \frac{\kappa}{We} \right) \mathbf{n} dS = 0. \quad (3.23)$$

This is a surface integral of the dynamic condition (3.11). However, since the width of the control volume can be chosen arbitrarily, the integrand has to vanish everywhere on the interface, which is equivalent to Eq. (3.11).

3.2 Coordinate Systems

It is possible to take advantage of the problem geometry by choosing the appropriate coordinate system. In this thesis we will only consider two-dimensional Cartesian and axisymmetric problems. Therefore, Cartesian and cylindrical coordinates are introduced in the present section. For Cartesian coordinates (x, y) , the nabla operator ∇ has the usual definition:

$$\nabla = \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix}. \quad (3.24)$$

The Cartesian form of the governing equations can simply be obtained by using Eq. (3.24) in Eq. (3.17).

In cylindrical coordinates, restricting to axisymmetric flow without swirl, the Navier-Stokes equations read as

$$\rho \frac{D\mathbf{v}}{Dt} = - \begin{pmatrix} \partial_x p \\ \partial_r p \end{pmatrix} + \frac{\mu}{Re} \frac{1}{r} \begin{pmatrix} \partial_r (r(\partial_r u + \partial_x v)) + 2r\partial_{xx} u \\ 2\partial_r (r\partial_r v) + r\partial_x (\partial_r u + \partial_x v) - 2v\frac{1}{r} \end{pmatrix} + \rho \frac{\mathbf{g}}{Fr^2} + \frac{1}{We} \mathbf{f}_{st}, \quad (3.25)$$

$$\partial_x u + \partial_r v + \frac{v}{r} = 0,$$

where the velocity vector $\mathbf{v} = (u, v)^T$ is expressed in terms of its components in axial (x) and radial (r) coordinate directions, respectively. For the derivation of the cylindrical Navier-Stokes equations the reader is referred to appendix A.

In the axisymmetric case, equations (3.25) are solved on a domain Ω , which is the meridian plane of a 3D cylindrical domain, see Fig. 3.1. The geometry and the flow variables are assumed

to be independent of the rotation angle ϕ . Therefore, a symmetry boundary condition of

$$v = 0 \quad \text{and} \quad \frac{\partial u}{\partial r} = 0$$

is required on the symmetry axis, i.e. for $(x, r = 0)$.

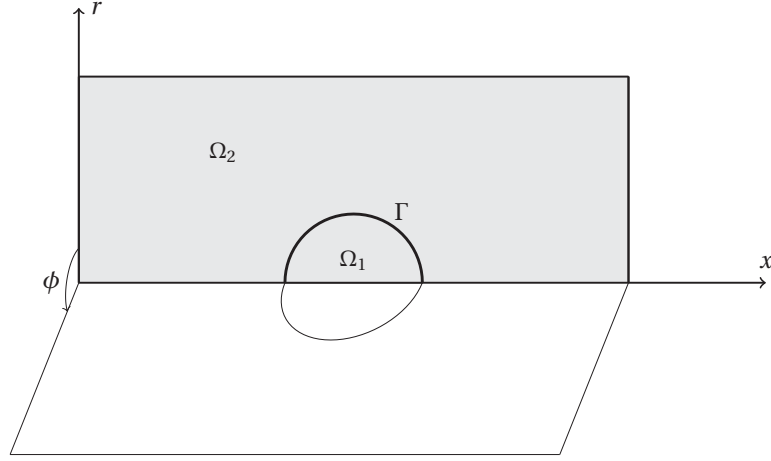


Figure 3.1: Meridian plane Ω of the axisymmetric cylindrical domain.

3.2.1 Axisymmetric Curvature

The curvature of a general axisymmetric surface can be obtained using a surface defined as a level surface of a function $F(x, y, z)$. Then we have:

$$\mathbf{n} = \frac{\nabla F}{|\nabla F|}$$

for the normal vector. The surface is defined by $F(x, y, z) = 0$. Using $F(x, y, z) = y^2 + z^2 - r^2(x)$, which describes any surface axisymmetric around the x -axis, the curvature can be computed by gives

$$\kappa = -\nabla \cdot \mathbf{n} = \frac{\frac{d^2 r}{dx^2}}{\left(1 + \left(\frac{dr}{dx}\right)^2\right)^{3/2}} - \frac{1}{r \left(1 + \left(\frac{dr}{dx}\right)^2\right)^{1/2}}. \quad (3.26)$$

The first term in Eq. (3.26) is the curvature of the curve $y = r(x)$ that is the intersection of the surface with a plane including the x -axis. The second term in Eq. (3.26) can be rewritten as [101]

$$-\frac{1}{r \left(1 + \left(\frac{dr}{dx}\right)^2\right)^{1/2}} = \frac{\sin(\psi)}{r},$$

where the angle ψ is defined in Fig. 3.2. It is the angle between the interface normal vector and the symmetry axis. The curvature of an axisymmetric surface can thus be written as

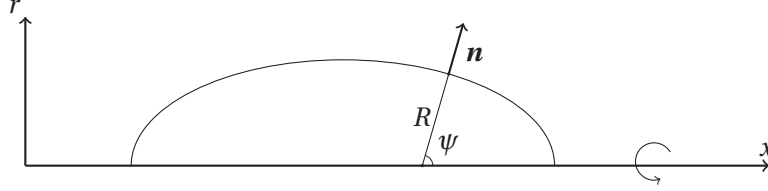


Figure 3.2: Definitions related to the axisymmetric curvature calculation.

$$\kappa = \bar{\kappa} + \tilde{\kappa}, \quad (3.27)$$

where $\bar{\kappa}$ is the curvature in the x - r plane and $\tilde{\kappa}$ is the axisymmetric component given by

$$\tilde{\kappa} = \frac{\sin(\psi)}{r} = \frac{1}{R}, \quad (3.28)$$

where R is the signed distance from the interface point to the point where the normal intersects the symmetry axis, with the sign being positive if the normal points away from the symmetry axis and negative if the normal points toward the symmetry axis.

3.3 ALE Formulation

The governing equations of fluid mechanics can be formulated using either of the two classical kinematical descriptions of motion being the Lagrangian and the Eulerian one. The Eulerian description is based on a fixed reference frame and the Lagrangian description follows the material particles. Since it is usually impracticable to follow the individual particles in a flow, the Eulerian formulation is more frequently used in fluid mechanics. However, the Eulerian formulation is less suited when the boundaries of the flow domain evolve with time. Combining the Lagrangian and the Eulerian descriptions becomes interesting for such cases. In this section the arbitrary Lagrangian-Eulerian (ALE) formulation, which underlies the present moving mesh method is described. The ALE approach combines the advantages of Lagrangian and Eulerian description, while minimizing their respective drawbacks. It allows to handle moving boundaries and interfaces without requiring to follow the individual fluid particles in Lagrangian manner.

In the Eulerian framework, the material time derivative (D/Dt) is defined by Eq. (3.7), where $\partial/\partial t$ is the local time derivative at a point in the Eulerian coordinates frame and ∇ is the derivative with respect to the Eulerian coordinates system. The Eulerian frame and its coordinates are assumed to be time-invariant. One can express the material derivative with respect to a

time dependent reference frame \mathcal{X} , that is attached to a an evolving domain, as [36]

$$\begin{aligned} \frac{D\bullet}{Dt} &= \left. \frac{\partial \bullet}{\partial t} \right|_{\mathcal{X}} + \mathbf{c} \cdot \nabla \bullet, \\ \mathbf{c} &= \mathbf{v} - \hat{\mathbf{v}}. \end{aligned} \quad (3.29)$$

The first difference between Eq. (3.7) and Eq. (3.29) is the local time derivative, where in Eq. (3.29) it represents the derivative at a fixed location in a coordinates system of \mathcal{X} , which is now allowed to change in time. The second difference is the velocity present in the convective term, where \mathbf{v} is the fluid velocity and $\hat{\mathbf{v}}$ is the velocity of the points of the coordinates system \mathcal{X} . All velocities are expressed with respect to the Eulerian frame.

The Lagrangian or material derivative is recovered from the ALE derivative if \mathcal{X} is chosen to be the material frame. That is is every point of \mathcal{X} corresponds to a material point and thus moves with the fluid velocity $\hat{\mathbf{v}} = \mathbf{v}$ hence we have

$$\frac{D\bullet}{Dt} = \left. \frac{\partial \bullet}{\partial t} \right|_{\mathcal{X}}.$$

Another limiting case is when the reference configuration \mathcal{X} is taken to be time independent, which corresponds to the Eulerian description. Then $\hat{\mathbf{v}} = \mathbf{0}$ and Eq. (3.29) reduces to Eq. (3.7). Therefore, the Lagrangian and the Eulerian derivatives are both limiting cases of Eq. (3.29). In a moving mesh algorithm, $\hat{\mathbf{v}}$ is the mesh velocity and \mathcal{X} is the reference frame defined by the mesh.

3.4 Conservation of Energy

Conservation of energy requires that the rate of change of energy in a parcel of fluid be balanced by the work done by surface and body forces and the heat flowing from the boundary. The total energy per unit mass is a sum of macroscopic kinetic energy $\mathbf{v}^2/2 = (\mathbf{v} \cdot \mathbf{v})/2$ and internal energy E . Neglecting any type of volumetric heat sources, heat radiation and assuming incompressible flow, the conservation of energy, expressed in terms of dimensional quantities, reads as:

$$\rho \frac{D(E + \mathbf{v}^2/2)}{Dt} = \nabla \cdot (\lambda \nabla T - \mathbf{v}p + \mathbf{v} \cdot [\mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T)]) + \rho \mathbf{v} \cdot \mathbf{g}, \quad (3.30)$$

where the left hand side is the material derivative of the sum of kinetic and internal energy. The first term on the right hand side represents the conductive transfer of heat, described by Fourier's law, with the thermal conductivity λ . The second and third term on the right hand side of Eq. (3.30) represent the work done by surface forces and the last term is the work of the gravitational volume force. Scalar multiplication of the momentum equation (3.5), in

Chapter 3. Governing Equations

dimensional form, with \mathbf{v} gives

$$\rho \frac{D(\mathbf{v}^2/2)}{Dt} = \mathbf{v} \cdot (-\nabla p + \nabla \cdot [\mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T)] + \rho \mathbf{g}) \quad (3.31)$$

$$= \nabla \cdot (-\mathbf{v} p + \mathbf{v} \cdot [\mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T)]) - \nabla \mathbf{v} : [\mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T)] + \rho(\mathbf{v} \cdot \mathbf{g}), \quad (3.32)$$

where in the last step use has been made of $\nabla \cdot \mathbf{v} = 0$. Subtracting the last equation from (3.30) gives

$$\rho \frac{DE}{Dt} = \nabla \cdot (\lambda \nabla T) + \nabla \mathbf{v} : [\mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T)], \quad (3.33)$$

where the last term on the RHS is the viscous dissipation. It is commonly assumed that the internal energy E is a function of temperature only and for an incompressible fluid $E = c_p T$, where c_p is the specific heat capacity.

In contrast to the previous sections, the energy equation above has been expressed in dimensional form. The non-dimensional form of the energy equation is obtained when the dimensional quantities (subscript *dim*) are made dimensionless as follows

$$T = \frac{T_{dim} - T_{sat}}{T_0}, \quad c_p = \frac{(c_p)_{dim}}{(c_p)_l}, \quad \lambda = \frac{\lambda_{dim}}{\lambda_l}, \quad (3.34)$$

where T_0 is a reference temperature and T_{sat} is the saturation temperature. The non-dimensional form the energy equation, is thus given as:

$$\rho c_p \frac{DT}{Dt} = \frac{\nabla \cdot (\lambda \nabla T)}{Re Pr} + \frac{Ec}{Re} \nabla \mathbf{v} : [\mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T)],$$

where Pr is the **Prandtl number**:

$$Pr = \frac{\mu_l (c_p)_l}{\lambda_l},$$

and the **Eckert number** is defined as

$$Ec = \frac{V^2}{(c_p)_l T_0}.$$

Here we will assume that the characteristic velocity V is sufficiently small for $Ec \ll Re$ to hold and viscous dissipation will thus be neglected. With this further assumption the energy equation reduces to:

$$\rho c_p \frac{DT}{Dt} = \frac{\nabla \cdot (\lambda \nabla T)}{Re Pr}. \quad (3.35)$$

Interface matching conditions are needed for the temperature field. The integral energy conservation principle around the interface yields the energy jump condition:

$$\mathbf{n} \cdot [\lambda \nabla T]_{\Gamma} = 0, \quad (3.36)$$

where we have neglected changes in surface energy, work done by viscous stresses and no phase change has been assumed. Equation (3.36) requires that the normal conductive heat flux be continuous at interface. A second interface condition can be deduced from a similar argument to the one used for the tangential velocity component. Namely that molecular interactions are expected to remove any discontinuity in the temperature field [5]. Therefore, we assume that the temperature is continuous across the fluids phases: $T_l = T_v$.

3.5 Phase Change

When phase change occurs there will be a mass transfer from the liquid to the vapor phase for boiling or vice-versa for condensation. Here we assume that phase change can only happen at the interface and no nucleation is allowed. A mass flow rate per unit interfacial area \dot{m}'' accounts for phase change in the general form of the interface jump conditions. Letting \mathbf{v}_l and \mathbf{v}_v denote the fluid velocity in the liquid and gas phase respectively and \mathbf{v}_{Γ} the interface velocity, the mass balance condition (3.8) reads as

$$\rho_l(v_l^n - v_{\Gamma}^n) = \rho_v(v_v^n - v_{\Gamma}^n) = \dot{m}'', \quad (3.37)$$

where $v^n := \mathbf{n} \cdot \mathbf{v}(\mathbf{x})$ denotes the normal velocity component and \mathbf{n} is the interface unit normal vector. When phase change occurs ($\dot{m}'' \neq 0$), the interface mass balance condition (3.37) will not reduce to Eq. (3.9) and the normal component of the velocity will differ if the densities differ, hence leading to a discontinuous velocity field. Rearranging Eq. (3.37), the normal velocity jump across the interface is obtained as

$$v_l^n - v_v^n = \left(\frac{1}{\rho_l} - \frac{1}{\rho_v} \right) \dot{m}''. \quad (3.38)$$

Two equations for the interface (normal) velocity can be obtained from Eq. (3.37):

$$v_{\Gamma}^n = \frac{\rho_l v_l^n + \rho_v v_v^n - 2\dot{m}''}{\rho_l + \rho_v},$$

and

$$v_{\Gamma}^n = \frac{v_l^n + v_v^n}{2} - \frac{\dot{m}''}{2} \left(\frac{1}{\rho_l} + \frac{1}{\rho_v} \right). \quad (3.39)$$

In the presence of phase change, the momentum jump condition reads

$$\dot{m}'' [\mathbf{v}]_{\Gamma} + \mathbf{n}[Ip - \frac{\mu}{Re}(\nabla \mathbf{v} + \nabla \mathbf{v}^T)]_{\Gamma} + \frac{\kappa}{We} \mathbf{n} = \mathbf{0}, \quad (3.40)$$

Chapter 3. Governing Equations

and assuming kinetic energies and viscous stresses are small, the energy jump condition is given by

$$\dot{m}'' \Delta h_{lv} - \frac{\mathbf{n}}{RePr} \cdot [\lambda \nabla T]_{\Gamma} = 0, \quad (3.41)$$

where Δh_{lv} is the latent heat of vaporization and the dimensional quantities (denoted by a subscript *dim*) have been non-dimensionalized as follows

$$\Delta h_{lv} = \frac{\Delta(h_{lv})_{dim}}{(c_p)_l T_0}, \quad \dot{m}'' = \frac{\dot{m}''_{dim}}{\rho_l U}. \quad (3.42)$$

Equations (3.40) and (3.41) reduce to (3.11) and (3.36) respectively when no phase change occurs.

Phase change has introduced an additional unknown, namely the evaporative mass flux \dot{m}'' . An additional interface condition is thus needed. However, such a condition cannot be derived from the macroscopic conservation principles and its choice warrants some discussion [71, 104, 105]. In this study, the interface temperature is assumed to be always equal to the saturation temperature. Combining this assumption with the continuity of temperature field, the interface temperature equals the temperature of each phase at the interface:

$$T_l = T_v = T_{sat}. \quad (3.43)$$

This condition can also be deduced from vanishing interface entropy production arguments [71] but it does not account for physical effects like the variation of saturation temperatures due to the pressure jump and the interface thermal resistance due to molecular effects. However, Eq. (3.43) is a good approximation for many situations of practical interest [71, 104], with a few exceptions like liquid metals. Here we assume that when the interface reaches the saturation temperature T_{sat} , heat flowing to the interface is not used to change the local temperature but to initiate a phase transition. The energy balance condition, expressed by Eq. (3.41), can be used to compute the mass flux \dot{m}'' as:

$$\dot{m}'' = \frac{\mathbf{n} \cdot [\lambda \nabla T]_{\Gamma}}{RePr \Delta h_{lv}}. \quad (3.44)$$

The interface jump conditions for mass (3.37), momentum (3.40) and energy (3.41), including phase change, can equivalently be included as singular source terms in the 'one fluid' framework. The latter approach has already been used for surface tension, which has been included as a singular volume force in the momentum equation (3.17). Applying the 'one fluid' formulation to the interface mass balance (3.37) gives a modified continuity equation:

$$\nabla \cdot \mathbf{v} = \left(\frac{1}{\rho_l} - \frac{1}{\rho_v} \right) \dot{m}'' \delta, \quad (3.45)$$

where δ is the Dirac delta. The effects of phase change on the velocity field, i.e. the discontinuous velocity given by Eq. (3.38), are thus accounted for by the right hand side of Eq. (3.45) without imposing (3.37) explicitly at the interface. Since the energy balance condition is enforced explicitly by computing \dot{m}'' from Eq. (3.44), the 'one fluid' framework is not used for the energy equation and Eq. (3.35) thus remains unchanged.

4 Numerical Method

4.1 Finite Element Method

The finite element method (FEM) is a valuable tool for solving partial differential equations (PDE) in engineering ranging from linear elasticity to fluid dynamics and heat transfer. A good description of FEM can be found in classical numerical analysis textbooks [108, 30] or in [35, 149] with an emphasis on fluid mechanics. In the FEM the domain is partitioned into a mesh of discrete elements and a weighted residual formulation is used to obtain a system of matrix equations yielding an approximate solution of the differential equation. The weighted residual formulation or variational formulation of a PDE is obtained by multiplying the governing equation with test functions and integrating over the domain. It is a weak form of the PDE, whose solution is not required to hold in a 'pointwise sense' i.e. everywhere in the domain. The resulting integral, which is called the residuum, is required to vanish for all test functions from a function space \mathcal{T} . In the FEM context the solution is expanded in a basis of the function space \mathcal{S} called shape function space, where in general \mathcal{S} and \mathcal{T} can be different. The functions spanning \mathcal{S} are defined locally over elements of the mesh and are commonly taken to be piecewise polynomial functions. In the present work the mesh is a subdivision of a region of the plane into triangular elements, which meet only at shared edges and vertices. A system of algebraic equations is obtained when both spaces \mathcal{S}, \mathcal{T} are taken to be finite dimensional vector spaces. For the Galerkin approach, used in the present context, the test function space and the shape function space are the same.

The Galerkin method has been very successful in solving linear PDE in structural mechanics and heat conduction because it can be shown that the solution possesses a 'best approximation' property. That is the difference between the FEM solution and the exact solution is minimized with respect to a certain norm. Unfortunately, two difficulties are associated with the numerical solution of the incompressible Navier-Stokes equations by the Galerkin approach. The first is related to the discretization of the non-linear convective term and manifests itself via spurious velocity oscillations in high Reynolds number flow when boundary layers are present. This issue can be cured by stabilized formulations, which add integral terms

to the weak form, such as the streamline-upwind Petrov-Galerkin (SUPG) [22] or Galerkin Least Squares (GLS) [68]. The second difficulty is related to the nature of the incompressible flow equations, where the pressure acts as a Lagrange multiplier of the incompressibility constraint [35], and requires an appropriate combination of interpolation functions for the velocity and the pressure.

4.1.1 Variational Formulation of the Governing Equations

In this section the Galerkin method is applied to the Navier-Stokes equations (3.17). We first consider the equations in Cartesian coordinates, for which the weak form is given in [35] and subsequently give the modifications required for axisymmetric equations in cylindrical coordinates. Using the FEM in space and time is possible, see [88, 135] but not pursued in this work, where the FEM is only applied to the spatial problem. The weak formulation of the Navier-Stokes equations is obtained by projecting the equations (3.17) onto a space of weighting functions \mathbf{w} for the velocity $\mathbf{v}(\mathbf{x}, t)$ and q for the pressure $p(\mathbf{x}, t)$. This results in the following weak form: for every $t \in [t_0, t_e]$, find $u(\bullet, t) \in \mathcal{U}_{u_D}$, $v(\bullet, t) \in \mathcal{V}_{v_D}$ and $p(\bullet, t) \in \mathcal{Q}$ such that for all $(\mathbf{w}, q) \in \mathcal{U}_0 \times \mathcal{V}_0 \times \mathcal{Q}$:

$$\begin{aligned} M\left(\rho \frac{D\mathbf{v}}{Dt}, \mathbf{w}\right) + \frac{1}{Re} K(\mathbf{v}, \mathbf{w}) + G(p, \mathbf{w}) &= \frac{1}{Fr^2} M(\rho \mathbf{g}, \mathbf{w}) + \frac{1}{We} G(H, \kappa \mathbf{w}), \\ D(q, \mathbf{v}) &= 0, \\ \text{with } \mathbf{v}(\mathbf{x}, t = t_0) &= \mathbf{v}_0(\mathbf{x}). \end{aligned} \quad (4.1)$$

Here, $\mathcal{U}, \mathcal{V}, \mathcal{Q}$ are the function spaces for the velocity components (u, v) and the pressure respectively. The following bilinear forms have been introduced in Eq. (4.1):

$$\begin{aligned} M(\mathbf{v}, \mathbf{w}) &= \int_{\Omega} \mathbf{w} \cdot \mathbf{v} \, d\mathbf{x}, \\ K(\mathbf{v}, \mathbf{w}) &= 2 \int_{\Omega} \mu \mathcal{D}(\mathbf{w}) : \mathcal{D}(\mathbf{v}) \, d\mathbf{x}, \\ G(q, \mathbf{w}) &= \int_{\Omega} \nabla q \cdot \mathbf{w} \, d\mathbf{x}, \\ D(q, \mathbf{w}) &= G(q, \mathbf{w}), \end{aligned} \quad (4.2)$$

where $2\mathcal{D}(\mathbf{v}) = \nabla \mathbf{v} + \nabla \mathbf{v}^T$.

Integrating the viscous stress tensor $\bar{\bar{\tau}}_v = 2\mu \mathcal{D}(\mathbf{v})$ by parts allows for transfer of the derivatives to the test function \mathbf{w} . No boundary term appears because we consider only homogeneous Neumann boundary conditions for the viscous stress tensor. Therefore, the boundary $\partial\Omega$ of Ω is composed of two distinct regions $\partial\Omega_D, \partial\Omega_N$ on which Dirichlet and Neumann boundary

conditions hold respectively:

$$\begin{aligned} \mathbf{v} &= \mathbf{v}_D \quad \text{on} \quad \partial\Omega_D, \\ \bar{\tau}_v \mathbf{n} &= 0 \quad \text{on} \quad \partial\Omega_N. \end{aligned}$$

The function spaces \mathcal{U}_{u_D} , \mathcal{V}_{v_D} and \mathcal{Q} are defined as

$$\begin{aligned} \mathcal{U}_{u_D} &:= \{ u \in H^1(\Omega) \mid u = u_D \quad \text{on} \quad \partial\Omega_D \}, \\ \mathcal{V}_{v_D} &:= \{ v \in H^1(\Omega) \mid v = v_D \quad \text{on} \quad \partial\Omega_D \}, \\ \mathcal{Q} &:= \{ q \in L^2(\Omega) \mid \int_{\Omega} q \, d\mathbf{x} = 0 \}. \end{aligned}$$

Here $H^1(\Omega)$ is a Sobolev space consisting of functions which are square-integrable and have square integrable first derivatives. The space of functions that are square integrable over the domain Ω is defined as:

$$L^2(\Omega) := \{ f \mid \int_{\Omega} f^2 \, d\mathbf{x} < \infty \}.$$

Taking the pressure function space with zero average $\int_{\Omega} p \, d\mathbf{x} = 0$ is a way to get a unique solution since only derivatives of the pressure appear in the equations making pressure defined up to an arbitrary constant. What is done numerically instead of requiring the pressure to have a zero average is to fix the pressure value at an arbitrary point of the domain.

The Galerkin method applied to the temperature equation (3.35) gives the weak form for the temperature field: for every $t \in [t_0, t_e]$, find $T(\bullet, t) \in \mathcal{U}_{T_D}$ such that for all $\theta \in \mathcal{U}_0$:

$$M\left(\rho c_p \frac{DT}{Dt}, \theta\right) + \frac{1}{RePr} L(\theta, T) = 0, \quad (4.3)$$

where $T(\mathbf{x}, t = t_0) = T_0(\mathbf{x})$ and the following bilinear form has been used:

$$L(\theta, T) = \int_{\Omega} \lambda \nabla \theta \cdot \nabla T \, d\mathbf{x}.$$

Axisymmetric Case

The axisymmetric case differs slightly from the Cartesian one and requires the introduction of so called weighted function spaces [15, 12, 34]. Consider the space of weighted square

integrable functions $L_\alpha^2(\Omega)$ for any $\alpha \in \mathbb{Z}$ to be composed of functions w for which

$$\int_{\Omega} w^2 r^\alpha d\mathbf{x} < \infty,$$

where $d\mathbf{x} = dx dr$. The subspace \mathcal{Q} of $L_1^2(\Omega)$ is defined as

$$\mathcal{Q} := \{ q \in L_1^2(\Omega) \mid \int_{\Omega} q r d\mathbf{x} = 0 \}.$$

Furthermore, consider the space $H_1^1(\Omega)$ as being the space of functions in $L_1^2(\Omega)$ such that their first partial derivatives are also in $L_1^2(\Omega)$. The spaces $\mathcal{U}_{u_D}, \mathcal{V}_{v_D}$ are defined as

$$\begin{aligned} \mathcal{U}_{u_D} &:= \{ u \in H_1^1(\Omega) \mid u = u_D \text{ on } \Gamma_D \}, \\ \mathcal{V}_{v_D} &:= \{ v \in H_1^1(\Omega) \mid v \in L_{-1}^2(\Omega) \text{ and } v = v_D \text{ on } \Gamma_D \}. \end{aligned}$$

The axisymmetric bilinear forms are given by:

$$\begin{aligned} M(\mathbf{v}, \mathbf{w}) &= \int_{\Omega} \mathbf{w} \cdot \mathbf{v} r d\mathbf{x}, \\ K(\mathbf{v}, \mathbf{w}) &= 2 \int_{\Omega} \mu \mathcal{D}(\mathbf{w}) : \mathcal{D}(\mathbf{v}) r d\mathbf{x} + 2 \int_{\Omega} \mu \frac{w_2 v}{r} d\mathbf{x}, \\ G(q, \mathbf{w}) &= \int_{\Omega} \nabla q \cdot \mathbf{w} r d\mathbf{x}, \\ D(q, \mathbf{v}) &= \int_{\Omega} q \nabla \cdot \mathbf{v} r d\mathbf{x} + \int_{\Omega} q v d\mathbf{x}, \\ L(\theta, T) &= \int_{\Omega} \lambda \nabla \theta \cdot \nabla T r d\mathbf{x}. \end{aligned} \tag{4.4}$$

Here, $\nabla = (\partial_x, \partial_r)^T$ has been used, where x is the axial and r the radial coordinate.

4.1.2 Semi Discrete System

The next step in the FEM is to go from the continuous weak form to a discrete system. Therefore, the solution is sought in finite dimensional subspaces of $\mathcal{U}_{u_D}, \mathcal{V}_{v_D}, \mathcal{Q}$, which are denoted $\mathcal{U}_{u_D}^h, \mathcal{V}_{v_D}^h, \mathcal{Q}^h$ and are composed of continuous piecewise-polynomial functions. The spatial domain Ω is subdivided into finite elements and the discrete solution \mathbf{v}^h, p^h is expressed in

terms of shape functions and associated nodal values:

$$\mathbf{v}^h(\mathbf{x}) = \sum_i^{N_v} \hat{\phi}_i(\mathbf{x}) \mathbf{v}_i, \quad p^h(\mathbf{x}) = \sum_i^{N_p} \hat{\psi}_i(\mathbf{x}) p_i \quad (4.5)$$

where N_v, N_p are the number of velocity and pressure nodes respectively, $\hat{\phi}_i$ are velocity shape functions and $\hat{\psi}_i$ are the pressure shape functions. The interpolating functions $\hat{\phi}_i(\mathbf{x})$ and $\hat{\psi}_i(\mathbf{x})$ have a compact support and they span the function spaces $\mathcal{W}_{u_D}^h, \mathcal{V}_{v_D}^h$ and \mathcal{Q}^h respectively. Moreover, they satisfy the relationship:

$$\hat{\phi}_i(\mathbf{x}_j) = \hat{\psi}_i(\mathbf{x}_j) = \delta_{i,j},$$

where $\delta_{i,j}$ is the Kronecker-Delta. This means every shape function is zero at all but one node of the mesh where its value is one. The integrals in (4.4) are computed by means of a standard Gaussian quadrature rule, using 12 quadrature points, which is exact for polynomials of degree 6 or less [29]. Since no quadrature points are located on the element boundary the axisymmetric terms, containing $1/r$, do not get singular. Accuracy issues could arise for very small element sizes, since the quadrature points would get very close to $r = 0$, but this did not pose any problem in this work. When this representation is used in the weighted residual of the Navier-Stokes equations (4.1) a system of ordinary differential equations (ODE) in time is obtained for the nodal values \mathbf{v}_i, p_i .

4.1.3 LBB Compatibility Condition

A combination of velocity/pressure shape functions is usually denoted as $PnPm$, where n stands for the degree of the polynomial for velocity interpolation and m stands for the degree of the pressure polynomial. The numerical treatment of the saddle point problem arising from the variational formulation of the incompressible Navier-Stokes equations poses restrictions on the combination of interpolation spaces for velocity and pressure. To guarantee the stability of a mixed FEM the interpolation spaces for pressure and velocity must satisfy a compatibility condition called Ladyzhenskaya-Babuska-Brezzi (LBB) condition or inf-sup condition [35]. In particular, the LBB condition does not allow equal order interpolations for velocity and pressure. This restriction is independent of the Reynolds number and even applies to the zero Reynolds number limit i.e. the Stokes equations.

Several combinations of velocity/pressure shape functions have been demonstrated to fulfil the LBB condition such as the $P_1^{bubble}P_1$ element [4] and the triangular Taylor-Hood element ($P2P1$) [64]. Figure 4.1 displays their interpolation nodes for the pressure and velocity shape functions. Using velocity-pressure shape function pairs that do not satisfy the LBB condition can result in spurious pressure oscillations. The LBB condition can be circumvented by eliminating the pressure unknown like in penalty methods [35] or by using a different set of test functions (Petrov-Galerkin method) that add stabilization terms [67]. In this work only LBB compatible elements (displayed in Fig. 4.1) are used such that no stabilization is required.

For the $P_1^{bubble}P_1$ element, the pressure is interpolated by linear shape functions and the velocity by linear shape functions supplemented by a cubic bubble function, with the control point at the element centroid. The three linear shape functions are interpolatory at the edges of the element and the cubic bubble function is interpolatory at the centroid. Since only four degrees of freedom are used for a cubic polynomial, the shape functions of this element are incomplete. That is, they do not span the space of third order polynomials. This results in a reduced rate of convergence with respect to a fully cubic interpolation but gives an LBB stable element without having to introduce too many degrees of freedom. The $P_1^{bubble}P_1$ pair has been called mini element because it fulfils the LBB condition with minimal degrees of freedom per element [4]. Moreover, the bubble function of the mini element acts in a way like the streamline diffusion of the streamline-upwind Petrov-Galerkin method and was shown to help in stabilizing the advective operator [21] without recourse to any additional stabilization.

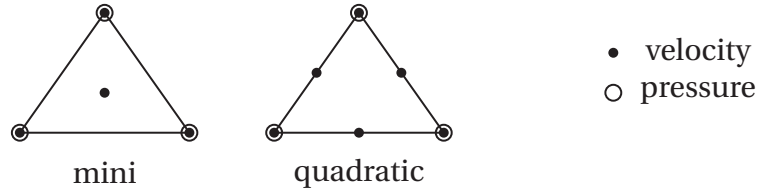


Figure 4.1: Interpolation nodes for the mini ($P_1^{bubble}P_1$) and quadratic (P_2P_1) element.

The quadratic element displayed in Fig. 4.1 interpolates the velocity with quadratic polynomials and the pressure is interpolated linearly. There are vertices in the middle of the triangle edges and as a result it has more degrees of freedom than the mini element. However, the triangular Taylor-Hood element exhibits quadratic convergence, while the convergence of the mini element is only linear. In the following, the shape functions for these two elements are presented.

4.1.4 Element Shape Functions

The shape functions are most conveniently defined in terms of barycentric coordinates. Barycentric coordinates are useful because they allow the shape functions to be defined as polynomials independent of the shape of the particular element. For a triangle there are three barycentric coordinates λ_{1-3} , each associated with one node of the triangle. The Cartesian coordinates of a point $\mathbf{x} = (x, y)^T$ located inside a triangle with nodes \mathbf{x}_{1-3} , are given by

$$\mathbf{x} = \sum_{i=1}^3 \lambda_i \mathbf{x}_i. \quad (4.6)$$

Barycentric coordinates are often called area coordinates because the barycentric coordinate corresponding to node i of the triangle is defined as the ratio of the area of the sub-triangle defined by the two other nodes and the point (x, y) to the triangle area A . By construction,

barycentric coordinates always satisfy:

$$\sum_{i=1}^3 \lambda_i = 1. \quad (4.7)$$

Since the pressure interpolation is linear, the pressure shape functions ψ_i are simply the barycentric coordinates:

$$\psi_i = \lambda_i, \quad i = 1-3.$$

The velocity shape functions ϕ_i of the mini element are:

$$\begin{aligned} \phi_i &= \lambda_i - 9 \prod_{j=1}^3 \lambda_j, \quad i = 1-3, \\ \phi_i &= 27 \prod_{j=1}^3 \lambda_j, \quad i = 4. \end{aligned}$$

For the quadratic Taylor-Hood element, the six shape functions are:

$$\begin{aligned} \phi_i &= 2\lambda_i(\lambda_i - 0.5), \quad i = 1-3, \\ \phi_i &= 4 \prod_{j=1, j \neq \hat{i}}^3 \lambda_j, \quad i = 4-6, \end{aligned}$$

where \hat{k} is one plus the remainder of the division of k by three:

$$\hat{k} = k \bmod 3 + 1.$$

4.2 Surface Tension Force

An accurate representation of the surface tension is essential to simulate capillary two-phase flow. In the FEM context, the surface tension term (3.19) can be computed by introducing the Laplace-Beltrami operator [39]. After multiplying with the weighting functions, the following term accounts for the effects of surface tension

$$\int_{\Omega} \mathbf{w} \mathbf{f}_{ST} d\mathbf{x} = \int_{\Omega} \mathbf{w} \kappa \mathbf{n} \delta d\mathbf{x}. \quad (4.8)$$

The definition of the Dirac distribution is such that

$$\int_{\Omega} \mathbf{w} \kappa \mathbf{n} \delta \, d\mathbf{x} = \int_{\Gamma} \mathbf{w} \kappa \mathbf{n} \, ds, \quad (4.9)$$

which transforms an integral over a volume containing the interface into a surface integral over the interface. Using the identity Eq. (1.3) to compute the curvature leads to the following expression for the surface tension term in the weak form

$$\int_{\Omega} \mathbf{w} \kappa \mathbf{n} \delta \, d\mathbf{x} = \int_{\Gamma} \mathbf{w} \nabla_S^2 i \, d\Gamma \, ds, \quad (4.10)$$

and then integration by parts can be used to get

$$\int_{\Gamma} \mathbf{w} \nabla_S^2 i \, d\Gamma \, ds = - \int_{\Gamma} \nabla_S \mathbf{w} : \nabla_S i \, d\Gamma \, ds + \int_{\partial\Gamma} \mathbf{w} \nabla_S i \, d\Gamma \, \mathbf{n} \, d\gamma. \quad (4.11)$$

The second term in Eq. (4.11) represents an integral over the boundary of the interface and has only to be considered when the interface is not a closed surface. The Laplace-Beltrami technique was used in [49] with a discrete version of Dirac distribution.

Another way to discretize the surface tension is to compute the curvature κ , which appears in the surface tension term, geometrically. This can be done by using polynomial curves to represent the interface [106] or NURBS [128] and then compute the curvatures of these known curves. If the interface is represented by straight element edges of a triangulation, then the curvature might be calculated efficiently using the first Frenet-Serret formula:

$$\bar{\kappa} \mathbf{n} = \frac{d\mathbf{t}}{ds}, \quad (4.12)$$

where \mathbf{t} is the interface unit tangent vector and s denotes the arc-length. The curvature given by Eq. (3.27) is a sum of the curvature $\bar{\kappa}$ inside the x - r -plane and the axisymmetric component $\tilde{\kappa}$. For the planar curvature, a finite difference version of the first Frenet-Serret formula is used:

$$\bar{\kappa} \mathbf{n} \approx \frac{\mathbf{t}_2 - \mathbf{t}_1}{\Delta s}. \quad (4.13)$$

Figure 4.2 illustrates the geometric quantities used for the curvature computation. Here the normal and tangent vectors \mathbf{n}_j , \mathbf{t}_j are defined on each interface segment and the curvature $\bar{\kappa}_i$ is defined at the interface vertices. The arc-length Δs in Eq. (4.13) is approximated as the arithmetic average of the lengths of the surrounding segments, see Fig. 4.2. The Frenet equation (4.13) gives the product of the curvature and interface unit normal vector ($\bar{\kappa} \mathbf{n}$) and the magnitude of the curvature is the Euclidian norm of this product. The axisymmetric

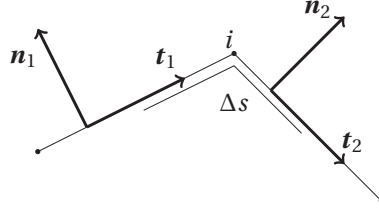


Figure 4.2: Discrete interface representation.

curvature component $\tilde{\kappa}$ is given by Eq. (3.28)

$$\tilde{\kappa}_i = \frac{\sin(\psi)}{r_i} = \frac{\hat{\mathbf{n}}_i \cdot \mathbf{e}_r}{r_i}, \quad (4.14)$$

where ψ is the angle that the interface normal forms with the x -axis and \mathbf{e}_r is the unit basis vector in the radial direction. The unit normal vectors at the nodes ($\hat{\mathbf{n}}_i$) are computed by normalizing the sum of the normal vectors of the two adjacent segments:

$$\hat{\mathbf{n}}_i = \frac{\mathbf{n}_1 + \mathbf{n}_2}{\|\mathbf{n}_1 + \mathbf{n}_2\|_2}. \quad (4.15)$$

Equation (4.14) can be used for all interface points outside the x -axis. On the x -axis, where $r_i = 0$, the rotational symmetry implies that the curvatures in the two normal directions are the same and therefore $\kappa_i = 2\tilde{\kappa}_i$ can be used.

Besides being very simple, equations (4.13) and (4.14) have some other advantageous properties. Equation (4.13) is a central difference approximation, which is second order accurate. As a result, decreasing the edge length of the segments that discretize the interface makes the curvature error decrease with the square of the edge length. Moreover, it can easily be demonstrated that in the particular case where uniformly spaced points are sampled on a half circle representing a spherical interface Eq. (4.13) and (4.14) give the exact curvatures of the circle and the sphere.

Using Eq. (3.20) with a discrete version of the heaviside function H , the normal vector and Dirac delta in Eq. (3.19) can be computed by taking the gradient of the discrete heaviside function. The discrete heaviside function H_h is defined such that it is equal to 1 for points in the inner phase, 0 for points in the other phase and 0.5 for interface points, see Fig. 4.3. The gradient of H_h is computed using the FEM with the linear shape functions also used for the pressure. Therefore, since the pressure and the heaviside function are both defined at the vertices of the elements and their gradient is computed in the same way, the discretization is consistent. That is, balance between the pressure gradient and the surface tension term in the momentum equation can be fulfilled up to numerical precision, as is going to be discussed in Section 5.4.

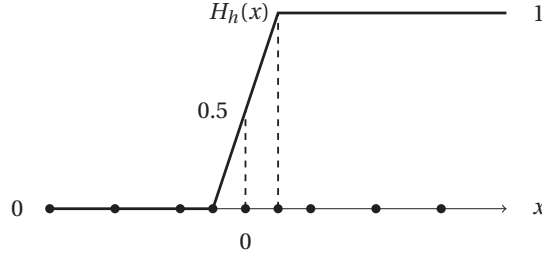


Figure 4.3: Discrete heaviside function.

4.3 Contact Angle

This section introduces a dynamic contact line model allowing to describe an interface moving along a solid boundary with a constant contact angle θ . We consider a liquid drop on a rigid surface with the configuration sketched in Fig. 4.4. Since the interface is defined by straight

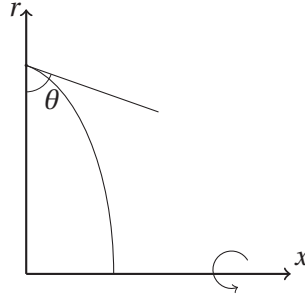


Figure 4.4: Schematic of the considered configuration: an axisymmetric liquid drop on a rigid surface defined by the plane $x = 0$.

line segments, a simple approximation is obtained by imposing the angle formed by the last interface segment with the plane $x = 0$. We obtain this by setting:

$$-\tan(\theta_s) = \frac{x_N - x_C}{r_N - r_C}, \quad (4.16)$$

where $(x, r)_C$ is the position of the contact point (the interface mesh vertex with $x = 0$) and $(x, r)_N$ the position of the interface vertex connected to $(x, r)_C$. In order to impose the desired contact angle we can simply require the geometry at the next time level to fulfil the above condition:

$$-\tan(\theta_s) = \left(\frac{x_N - x_C}{r_N - r_C} \right)^{n+1} = \frac{x_N^n - x_C^n + (\hat{u}_N - \hat{u}_C)\Delta t}{r_N^n - r_C^n + (\hat{v}_N - \hat{v}_C)\Delta t},$$

where $\hat{\mathbf{v}}$ is the mesh velocity, which determines the position of the mesh points at the new time level: $(x, r)^{n+1} = (x, r)^n + \hat{\mathbf{v}}\Delta t$. Here, we have $x_C = \hat{u}_C = 0$ since the contact point remains on the wall, which is assumed at $x = 0$. The contact angle θ_s can be imposed by choosing the

radial mesh velocity at the contact point as follows

$$\hat{v}_C = \hat{v}_N - \left(r_C^n - r_N^n - \frac{x_N^n + \hat{u}_N \Delta t}{\tan(\theta_s)} \right) \frac{1}{\Delta t}. \quad (4.17)$$

To impose the contact angle by Eq. (4.17), the kinematic condition is overridden at a single interface vertex. This is necessary otherwise the no slip condition would cause the contact point to remain pinned and no droplet spreading could be simulated. Although there is no physical justification for this procedure its effects are that dynamic wetting, with constant contact angle, can be simulated.

4.4 Phase Change

The numerical phase change model is based on the equations presented in section 3.5. As mentioned there, we assume a continuous temperature field with the vapor and liquid temperature being equal to the saturation temperature at the interface: $T_l = T_v = T_{sat}$. Therefore, we apply a uniform Dirichlet condition on the temperature by imposing $T = T_{sat}$ at all interface nodes. Now since the interface is a level curve of the temperature field, the interface normal vector and the gradient ∇T at the interface are parallel. The mass exchange between the phases is computed from Eq. (3.44). Based on Eq. (3.44), the evaporative mass flux (\dot{m}'') is proportional to the interfacial jump of the conductive heat flux $\mathbf{q} = \lambda \nabla T$:

$$\dot{m}'' \Delta h_{lv} = \frac{1}{Re Pr} [\mathbf{n} \cdot \mathbf{q}]_\Gamma. \quad (4.18)$$

The quantity \dot{m}'' enters the continuity equation (3.45) as a source term and defines the mesh displacement by Eq. (3.39). It is thus computed at every interface mesh node. However, since the temperature field is interpolated by piecewise linear shape functions, the temperature gradient is constant inside every mesh element but discontinuous at inter-element edges. In order to evaluate Eq. (4.18) at an interface node i , the one sided temperature gradient is computed using the area weighted average of the elements, containing node i , on either side of the interface. The value of the heat flux in every element e containing interface node i is used to compute the area average of the heat flux $\bar{\mathbf{q}}$ on the liquid and vapor side of the interface:

$$\bar{\mathbf{q}}_l = \frac{\sum_{e \in \Omega_l} \mathbf{q}_e A_e}{\sum_{e \in \Omega_l} A_e}, \quad \bar{\mathbf{q}}_v = \frac{\sum_{e \in \Omega_v} \mathbf{q}_e A_e}{\sum_{e \in \Omega_v} A_e},$$

where \mathbf{q}_e is the heat flux computed using the temperature gradient in element e with area A_e and Ω_j represents the set of neighboring elements in phase j (vapor or liquid). Note that the procedure has been described for planar meshes. However, this approach is easily extended to three dimensional unstructured meshes using a volume average instead of the area average.

The interfacial jump of the heat flux at node i is then obtained by $[\mathbf{q}]_i = \bar{\mathbf{q}}_l - \bar{\mathbf{q}}_v$. To obtain the normal derivative required in Eq. (4.18), the vector $[\mathbf{q}]_i$ must be multiplied by the interface normal vector \mathbf{n} at node i . An interface normal computed from the interface geometry could be used. However, to be consistent with the employed continuum surface tension method, the gradient of the heaviside function ($\delta \mathbf{n} = \nabla H$) is used here.

When phase change occurs, the weak form (4.1) has to be modified to include the velocity source term in the continuity equation. The following variational formulation is used for phase change problems: find $\mathbf{v} = (u, v) \in \mathcal{U}_{u_D} \times \mathcal{V}_{v_D}$, $p \in \mathcal{Q}$, $T \in \mathcal{U}_{T_D}$ such that

$$\begin{aligned} M\left(\rho \frac{D\mathbf{v}}{Dt}, \boldsymbol{\phi}\right) + \frac{1}{Re} K(\mathbf{v}, \boldsymbol{\phi}) + G(p, \boldsymbol{\phi}) &= \frac{1}{Fr^2} M(\rho \mathbf{g}, \boldsymbol{\phi}) + \frac{1}{We} G(H, \kappa \boldsymbol{\phi}), \\ D(\psi, \mathbf{v}) &= \left(\frac{1}{\rho_l} - \frac{1}{\rho_v}\right) \frac{G(H, [\mathbf{q}]\psi)}{Re Pr \Delta h_{lv}}, \\ M\left(\rho c_p \frac{DT}{Dt}, \theta\right) + \frac{1}{Re Pr} L(\theta, T) &= 0, \end{aligned} \quad (4.19)$$

for all $\boldsymbol{\phi} \in \mathcal{U}_0 \times \mathcal{V}_0$, $\psi \in \mathcal{Q}$, $\theta \in \mathcal{U}_0$. Here the function spaces and bilinear forms retain their previous definitions.

4.5 Semi-Lagrangian Time Discretization

In this work, a semi-Lagrangian method is used to discretize the time derivative. The semi-Lagrangian method was introduced in the middle of the last century and has been used extensively in the context of numerical weather prediction [127]. Its main advantage is that it is stable for larger time steps than Eulerian based advection schemes. This is because the semi-Lagrangian method discretizes the material time derivative and thereby avoids the stability issues stemming from the convective term. The idea behind the semi-Lagrangian method is to combine the best from both the Eulerian (fixed reference frame) and the Lagrangian (reference frame following a fluid particle) descriptions. This is achieved by using a fixed mesh and considering a special set of particles. The set of particles being chosen at each time step consists of particles that arrive precisely at the mesh points at the end of the time step.

The material time derivative is discretized by a finite difference approximation:

$$\frac{D\psi(\mathbf{x}, t)}{Dt} \approx \frac{\psi(\mathbf{x}, t + \Delta t) - \psi(\mathbf{x}^d, t)}{\Delta t}, \quad (4.20)$$

where ψ stands for a generic scalar or vector valued function. Here, $\psi(\mathbf{x}, t + \Delta t)$ is the value of ψ at the mesh point with position \mathbf{x} evaluated at the instant $t + \Delta t$. The semi-Lagrangian method is implemented by finding the departure point \mathbf{x}^d , i.e. the position at time t of the fluid particle located at \mathbf{x} at time $t + \Delta t$. Equation (4.20) is used to advance the unknown ψ in time, at every time level $t = t^n$, and $\psi(\mathbf{x}^d, t^n) = \psi^n(\mathbf{x}^d)$ is obtained through interpolation

of the known distribution ψ^n . The departure position is found by integrating the velocity backwards in time using an explicit Euler scheme:

$$\mathbf{x}^d = \mathbf{x} - \Delta t \mathbf{c}^n(\mathbf{x}), \quad (4.21)$$

where $\mathbf{c}^n(\mathbf{x})$ is the speed of point \mathbf{x} at time t^n . The relative position of a mesh point \mathbf{x}_i and its departure point \mathbf{x}^d is sketched in Fig. 4.5. In order to approximate the ALE derivative (3.29), the mesh motion needs to be included in the ALE velocity $\mathbf{c}^n = \mathbf{v}^n - \hat{\mathbf{v}}^n$, where $\hat{\mathbf{v}}^n$ is the mesh velocity. If the mesh is fixed then $\mathbf{c}^n = \mathbf{v}^n$.

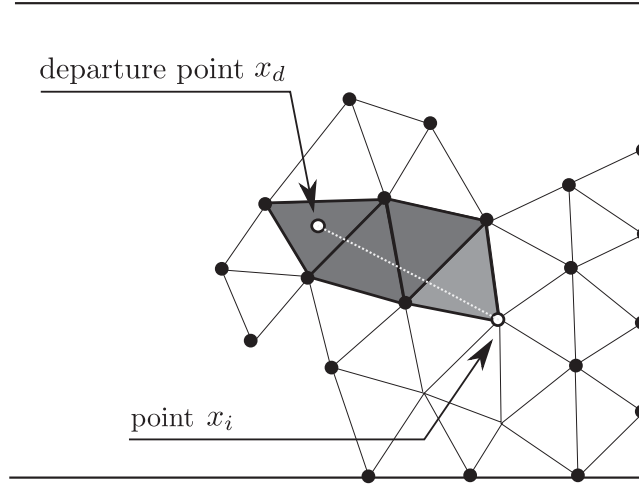


Figure 4.5: Departure point position.

The explicit Euler scheme used for the position of departure Eq. (4.21), where the velocity is taken as the velocity at the previous time level, is a first order approximation. A discussion of higher order approximations, for example a midpoint rule, can be found in [127]. Here, the accuracy obtained with Eq. (4.21) was found to be sufficient and it was thus preferred over more expensive higher order approximations. The semi-Lagrangian method is unconditionally stable with respect to the time step size [98]. This means that the time step size is not limited by the local advection speed, thus allowing to circumvent the Courant-Friedrichs-Lewy (CFL) condition [127]. In this work, the semi-Lagrangian approach was found to be stable for larger Δt in convection dominated problems but other effects like gravity or surface tension sometimes limited the time step size. Another advantage of the semi-Lagrangian method is that the resulting system of equations is linear¹, thus avoiding the need to resort to iterations or Newton-Raphson algorithms. The algorithmic difficulty of the semi-Lagrangian approach resides in the interpolation required to evaluate the value of ψ at the departure point $\psi(\mathbf{x}^d, t)$. Moreover, the first order finite differences approximations used in Eq. (4.20) is numerically dissipative compared to higher order schemes like Crank-Nicolson or Runge-Kutta methods.

¹This is due to the absence of the convective term, which is the only non-linear term in the Navier-Stokes equations.

4.6 Fully Discrete Equations

After discretizing both spatial and time coordinates, we obtain the following fully discrete variational form of the Navier-Stokes equations: find $(\mathbf{v}_h^{n+1}, p_h^{n+1}) \in \mathcal{U}_{u_D}^h \times \mathcal{V}_{v_D}^h \times \mathcal{Q}^h$ such that

$$\begin{aligned} \frac{M(\rho \mathbf{v}_h^{n+1}, \mathbf{w}_h)}{\Delta t} + \frac{K(\mathbf{v}_h^{n+1}, \mathbf{w}_h)}{Re} + G(p_h^{n+1}, \mathbf{w}_h) &= \frac{M(\rho \mathbf{v}_h^n(\mathbf{x}^d), \mathbf{w}_h)}{\Delta t} + \\ &\quad \frac{M(\rho \mathbf{g}_h, \mathbf{w}_h)}{Fr^2} + \frac{G(H_h, \kappa \mathbf{w}_h)}{We}, \\ D(q_h, \mathbf{v}_h^{n+1}) &= 0, \end{aligned} \quad (4.22)$$

for all $(\mathbf{w}_h, q_h) \in \mathcal{U}_0^h \times \mathcal{V}_0^h \times \mathcal{Q}^h$.

The integrals in (4.2) are evaluated element-wise, with the material properties ρ, μ taken to be constant in each element. In Eq. (4.22), the viscous term, the pressure gradient, the spatial operators and the interface dependent quantities (H_h, κ) are treated implicitly, i.e. they are evaluated for the mesh at the new time level $(n + 1)$. However, the position of the mesh points at time level $n + 1$ is obtained by displacing the points with the known velocity of time level n . Thus, resulting in a weak coupling of the flow field and interface geometry. An fully implicit treatment of the interface position could help relaxing the time step requirement for numerical stability imposed by the surface tension term. This could be achieved by displacing the mesh with the new velocity at time level $n + 1$. However, such an approach is computationally more expensive, probably requiring an iterative procedure and it was not found to be necessary for the computations presented in this thesis.

Equation (4.22) leads to an algebraic system that can be written as:

$$\begin{pmatrix} \mathbf{B} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U}^{n+1} \\ \mathbf{P}^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}, \quad (4.23)$$

where the unknown velocity components at all velocity nodes and the pressure unknowns at all pressure nodes are contained in the vectors $\mathbf{U}^{n+1} \in \mathbb{R}^{2N_v}$ and $\mathbf{P}^{n+1} \in \mathbb{R}^{N_p}$ respectively. The vectors $\mathbf{b}_1, \mathbf{b}_2$ contain the source terms, the known values at the previous time-level and the boundary conditions. This system is linear and can be solved with a direct or an iterative solver. Solving the system for velocity and pressure simultaneously can become computationally expensive for a large number of degrees of freedom. Therefore, it is often expedient to decouple the velocity and the pressure in order to reduce the system size. Different methods exist for this purpose. Here the so called algebraic splitting methods are going to be described.

4.6.1 Algebraic Splitting

Applying a block LU factorization to the system (4.23) yields

$$\begin{pmatrix} \mathbf{B} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{D} & -\mathbf{D}\mathbf{B}^{-1}\mathbf{G} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{B}^{-1}\mathbf{G} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}, \quad (4.24)$$

By first computing an intermediate velocity \mathbf{U}^* , system (4.23) can be solve in two steps:

$$\begin{aligned} \text{Step 1:} \quad & \mathbf{B}\mathbf{U}^* = \mathbf{b}_1, \\ & (\mathbf{D}\mathbf{B}^{-1}\mathbf{G})\mathbf{P}^{n+1} = \mathbf{D}\mathbf{U}^* - \mathbf{b}_2, \\ \text{Step 2:} \quad & \mathbf{U}^{n+1} = \mathbf{U}^* - \mathbf{B}^{-1}\mathbf{G}\mathbf{P}^{n+1}. \end{aligned}$$

These steps can even be simplified by eliminating the intermediate velocity:

$$\begin{aligned} (\mathbf{D}\mathbf{B}^{-1}\mathbf{G})\mathbf{P}^{n+1} &= \mathbf{D}\mathbf{B}^{-1}\mathbf{b}_1 - \mathbf{b}_2, \\ \mathbf{B}\mathbf{U}^{n+1} &= \mathbf{b}_1 - \mathbf{G}\mathbf{P}^{n+1}. \end{aligned}$$

However, this approach is impracticable for large problems, since it requires one to compute the inverse matrix \mathbf{B}^{-1} . An approximation of \mathbf{B}^{-1} is required for an efficient method. The idea is to replace the exact LU decomposition (4.24) by an approximate one [35]

$$\begin{pmatrix} \mathbf{B} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{pmatrix} \approx \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{D} & -\mathbf{D}\mathbf{H}_1\mathbf{G} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{H}_2\mathbf{G} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}, \quad (4.25)$$

Three families of methods can be distinguished. First, if $\mathbf{H}_1 \neq \mathbf{H}_2 \neq \mathbf{B}^{-1}$ both the momentum and mass conservation are perturbed. Second, using $\mathbf{H}_2 = \mathbf{B}^{-1}$ and $\mathbf{H}_1 \neq \mathbf{H}_2$ momentum is preserved but the continuity equation is modified. Third, if $\mathbf{H}_1 = \mathbf{H}_2 = \mathbf{H}$ but $\mathbf{H} \neq \mathbf{B}^{-1}$ mass conservation is granted but the momentum equation is modified. The last approach is the only one preserving mass. A common approach is to use a so called lumped matrix approximation, where the matrix \mathbf{H} is the inverse of the diagonal matrix obtained by summing all the entries of each row of \mathbf{B} into its diagonal component.

4.7 Boundary Conditions

The boundary $\partial\Omega$ of the computational domain Ω has distinct sub-regions where different boundary conditions (BC) hold. Dirichlet BC are assumed to hold on the $\partial\Omega_D$ part of the boundary, while on $\partial\Omega_N$ Neumann boundary conditions hold. Here, we only consider homo-

geneous Neumann boundary conditions for the viscous stress tensor $\bar{\bar{\tau}}_\nu$ such that

$$\mathbf{v} = \mathbf{v}_D \quad \text{on} \quad \partial\Omega_D, \quad \bar{\bar{\tau}}_\nu \mathbf{n} = \mathbf{0} \quad \text{on} \quad \partial\Omega_N.$$

Homogeneous Neumann boundary conditions are imposed naturally with the FEM by simply omitting the terms resulting from integration by parts. As a result the boundary term stemming from integration by parts vanishes and is excluded from the variational form (4.2). Dirichlet boundary conditions are imposed directly in the system matrix of (4.23) by setting the diagonal element to one and the right hand side to the desired value.

A third kind of boundary condition are periodic BCs, which are implemented in the following way. Periodicity is assumed in the x -direction on a domain with straight and vertical left and right boundaries. Identical boundary meshes are created initially on the boundaries, where the nodes of each periodic boundary are overlapping with the nodes of the other periodic boundary and offset by the length of the domain in x -direction. Therefore, each node of the right boundary (R) has a corresponding node on the left boundary (L). To enforce periodicity of the flow the dependent variables of the nodes on the right and left boundaries are connected as if the domain was folded onto itself. The first requirement is that the unknowns have the same value on the L nodes as on the corresponding R nodes. This condition is enforced by imposing a Dirichlet BC for the nodes on the left boundary. Then all the contributions of the left boundary nodes iL are added to the equation of the corresponding right boundary nodes iR . In this way periodicity is fully accounted for, as if there was an exact copy of the domain patched on each side.

When solving the Navier-Stokes equations on periodic domains, the velocity field is periodic but the pressure field is usually not periodic, e.g. due to a pressure gradient. One can still use periodic BCs and incorporate the pressure gradient. Periodic boundary conditions for the velocity with a constant pressure gradient p_x are imposed in the linear system (4.23) by letting:

$$\begin{aligned} U_{iL} &= U_{iR} \\ P_{iL} &= P_{iR} + p_x L \\ B_{iR,j} &= B_{iR,j} + B_{iL,j} \\ G_{iR,j} &= G_{iR,j} + G_{iL,j} \\ D_{iR,j} &= D_{iR,j} + D_{iL,j} \\ b_{iR}^1 &= b_{iR}^1 + b_{iL}^1, \end{aligned}$$

where L is the domain length and iL , iR are the indices of a corresponding pair of nodes on the left and right boundaries.

4.7.1 Interface Boundary Conditions

When the interface is a closed shape no boundary conditions are needed. However, when the interface ends on a boundary, as it is the case for a contact line or a symmetry boundary, geometrical boundary conditions need to be specified at the point where the interface ends. These numerical boundary conditions are required for the curvature calculation via Eq. (4.13), which is a central difference approximation and thus requires bilateral information. For most types of boundaries allowed to intersect the interface, finding an appropriate interface boundary condition is straightforward. For a symmetric boundary a reflection of the interface points positions on the symmetry line is applied and the curvature is calculated using the reflected points at the boundary. Similarly for a periodic boundary, the interface points of the opposite periodic boundary are translated such as to serve as neighboring points for the curvature calculation. However, finding a general geometric boundary condition for the contact angle boundary described in section 4.3 is not straightforward. This is because only the slope of the interface is prescribed at the boundary and not the curvature (the derivative of the slope). In the particular case where the contact angle is $\pi/2$ a symmetry boundary condition is used. For an arbitrary contact angle, the interface can be approximated locally by a circle, i.e. assuming a constant curvature at the interface boundary. This later gives accurate results when surface tension effects are dominant and the interface shape is close to a spherical cap.

4.8 Extension of Interface Quantities

The curvature κ in the surface tension term and the quantity $[q]_i$ described in the previous section are quantities that are defined only at the interface. However, since a weighted volume integral formulation is used, these quantities have to be integrated over a volume element. This requires attributing a value of these quantities to mesh points outside the interface.

Since interface quantities can vary only along the interface, they have to be constant along lines leaving in a direction normal to the interface. It can be shown that the line joining an arbitrary point and its closest point on the interface is normal to the interface. Therefore, an orthogonal extension is used, where for every mesh node the interface quantity is taken as the value of its closest interface node. The extension F_e of an interface quantity F , is computed as:

$$F_e(\mathbf{x}) = F(\mathbf{x}_c), \quad (4.26)$$

where $\mathbf{x}_c \in \Gamma$ is the point on interface, which is closest to \mathbf{x} :

$$\mathbf{x} = \mathbf{x}_c + d(\mathbf{x}) \mathbf{n}(\mathbf{x}_c). \quad (4.27)$$

This procedure requires finding the closest interface point \mathbf{x}_c for every mesh point, which is done by finding the point with the minimal distance $d(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_c\|$ to \mathbf{x} .

4.9 Computational Mesh

This section deals with all procedures related to the computational mesh. We focus first on the way the position of the mesh points is updated and then on the remeshing procedure. In this thesis an unstructured triangular mesh is used. A disadvantage of an unstructured mesh compared to structured grids is that point coordinates and connectivity need to be stored. However, an unstructured mesh allows a much greater flexibility in fitting complicated geometries, which is a key requirement for two-phase moving mesh computations. A FEM mesh requires mesh elements of appropriate size, with shapes that are not too elongated (bounds on angles) and element size should not change too abruptly. If triangular elements are used the best shape is the equilateral triangle.

The software Triangle [122] is used to triangulate the mesh elements for all computations. Triangle is a mesh generator, which can generate Delaunay and constrained Delaunay triangulations. The Delaunay triangulation of a discrete set of points in the plane is the dual graph of their Voronoi diagram ². Some properties of the Delaunay triangulation are that no point is found inside the circumcircle of any triangle and the minimum angle of all triangles is maximized. The latter property of the Delaunay triangulation is desirable for FEM meshes where acute angles increase the discretization error. A file describing the initial problem geometry in the ".msh" format [52] is supplied as an input to the simulation program. The ".msh" file contains the meshed domain boundaries. This one-dimensional mesh is used by the Triangle software to generate the two-dimensional domain mesh. While the mesh moving scheme is designed to preserve mesh quality, it cannot in general prevent serious mesh degradation and tangled mesh elements can occur causing the simulation to crash. Tangling refers to the situation where (due to mesh deformation) mesh elements lose their required topological qualities, e.g. vertex crossing the opposite face of the triangle. Therefore, remeshing is used to prevent tangling and to make sure that the mesh is adequately refined in every region. Every time remeshing is applied, the Triangle software is used to generate a triangulation of the new set of points.

4.9.1 Mesh Displacement

Let us first consider how to update the location of the mesh nodes representing the interface. When there is no phase-change, the interface condition (3.9) requires the normal component of the interface velocity to equal the normal fluid velocity:

$$\mathbf{v}_\Gamma \cdot \mathbf{n} = \mathbf{v} \cdot \mathbf{n}.$$

The tangential interface velocity can be shown to be irrelevant for the evolution of the interface position, see Lemma 5.2.1 in [51]. Therefore, in the continuous problem, only the normal velocity component $\mathbf{v}_n := (\mathbf{v} \cdot \mathbf{n})\mathbf{n}$ needs to be taken into account for the displacement of the

²The Voronoi diagram of a set of points is a partition of the plane into regions of closest distance to any point in the specified set

interface. However, numerical results are found to depend strongly on whether the normal velocity or the full fluid velocity \mathbf{v} (including its tangential component) are used to displace the interface points. There are cases where numerical accuracy is enhanced by keeping the tangential component and cases where it is better to remove it completely. The former holds for pure translational motion and the latter for pure rotational motion. Here a parameter γ is used to control how much of the tangential component is used for the mesh motion. In the absence of phase-change, the mesh velocity $\hat{\mathbf{v}}_i$ of interface nodes \mathbf{x}_i is specified as follows:

$$\hat{\mathbf{v}}_i = \mathbf{v}(\mathbf{x}_i) - \gamma \mathbf{v}_t(\mathbf{x}_i), \quad \text{for } \mathbf{x}_i \in \Gamma. \quad (4.28)$$

where $\mathbf{v}_t = \mathbf{v} - \mathbf{v}_n$ is the tangential component of the fluid velocity and the parameter γ can vary in $[0, 1]$. For simulations including phase-change, Eq. (3.39) is used for the interface mesh velocity:

$$\hat{\mathbf{v}}_i = \mathbf{v}(\mathbf{x}_i) - \frac{\dot{m}_i''}{2} \left(\frac{1}{\rho_l} + \frac{1}{\rho_v} \right) \mathbf{n}_i, \quad \text{for } \mathbf{x}_i \in \Gamma, \quad (4.29)$$

where in the source term approach $\mathbf{v}(\mathbf{x}_i)$ is the average of the velocities in both phases.

While the interface mesh motion is determined by the physics, mesh motion in the bulk fluid regions can be chosen arbitrarily. We choose the bulk mesh motion with the aim to limit mesh distortion. In our scheme, the mesh nodes that are not on the interface are thus moved with the following velocity:

$$\hat{\mathbf{v}}_i = \beta_1 \mathbf{v}(\mathbf{x}_i) + \beta_2 \hat{\mathbf{v}}_i^a, \quad \text{for } \mathbf{x}_i \notin \Gamma, \quad (4.30)$$

where \mathbf{v} is the fluid velocity and $\hat{\mathbf{v}}_i^a$ is an average of the mesh velocities of the surrounding nodes. The user defined parameters $\beta_{1,2}$ can be tuned to preserve a good mesh quality in any particular situation. An exception are mesh nodes on the boundaries of the domain, which are not moved.

While the mesh moving scheme (4.30) can usually avoid mesh tangling, the mesh quality is in general reduced. To increase the quality of the mesh, Laplacian smoothing is successively applied to the volume mesh. Laplacian smoothing consists in moving each mesh point towards the barycenter of the surrounding mesh points [31]. Therefore, new nodes positions $\tilde{\mathbf{x}}_i$ are calculated as follows

$$\tilde{\mathbf{x}}_i = \frac{1}{N} \sum_j^N \mathbf{x}_j,$$

where N represents the number of neighbour points at point i . Since the mesh is unstructured N may vary strongly over the mesh. After mesh motion and smoothing, mesh points are added and removed in order to keep the desired resolution in any region of the flow. Moreover, remeshing can also prevent the overlapping of elements since for the new mesh, a new triangulation of the set of nodes is generated.

4.9.2 Mesh Refinement and Remeshing

Remeshing can be applied at every time iteration or after a specified number of time iterations. The insertion and deletion of mesh points is based on a comparison between the local actual edge length l and a desired edge length expressed by a length density function h . In order to keep a smooth distribution of h , its distribution is obtained by solving a Helmholtz equation of the form

$$k \nabla^2 h = h - \tilde{h}. \quad (4.31)$$

where k is a user defined parameter which specifies the diffusivity level. The auxiliary density function \tilde{h} , which needs not be smooth, is specified directly by expressing the desired edge length in each region based on threshold functions. For low values of k , h is close to \tilde{h} , while for high values of k , Eq. (4.31) tends toward the Laplace equation, which will smooth away any peaks or discontinuities. Equation (4.31) is solved using linear finite elements on the same mesh used to solve the flow equations. Moreover, since the system matrix resulting from Eq. (4.31) is symmetric and positive-definite the Conjugate Gradient method is used to solve the linear system. The prescription of an adequate \tilde{h} is problem dependent and rely heavily on one's intuition on the form of the solution. For example, thin gaps or films are resolved by considering the distance from the interface and from the wall and prescribing a denser mesh in those regions. A criterion which is used near the interface is to prescribe a shorter edge size where the curvature is high. This can be achieved by considering the local curvature radius $R_c = \kappa_{2d}^{-1}$ and requiring a circle with radius R_c to be resolved by a minimum number of points.

After Eq. (4.31) has been solved, a loop is performed over all edges of the mesh and if the edge is too long ($l > t_1 h$) a mesh point is inserted in the middle of the edge, while if $l < t_2 h$ then one of the points is removed. The parameters t_1 and t_2 are tolerances, which can be used to vary the frequency of the insertion and deletion operations. Typical values are $t_1 = 2$ and $t_2 = 0.5$. That is, a point is inserted/deleted when the edge is longer than twice/lower than half the target edge length. For interface edges, inserting a point in the middle of the edge leads to a curvature error at the inserted point. Therefore, the position of inserted interface points are corrected by moving the point on a circle with the average curvature of the surrounding points. Additionally, elements which are too small in area are tracked and removed from the mesh.

In the present code, the lower dimensional mesh representing the interface and the boundaries of the domain is saved separately from the interior mesh. While points can be inserted/deleted also on the boundary mesh, only the interior mesh is entirely regenerated. The remeshing of the interior of the domain is performed using the Triangle software [122]. The interior mesh points are passed to the Triangle software and Triangle generates the new element connectivity arrays subject to the boundary conditions that the edges of the boundary elements match the boundary mesh.

4.10 Interpolation

The semi-Lagrangian method described in section 4.5 requires an interpolation to evaluate the departure points values $\psi(\mathbf{x}^d)$. When remeshing is used an interpolation is also required to obtain the values at target mesh nodes. A consistent Lagrangian interpolation is used in both cases. The value of a function ψ at an arbitrary interpolation point $\hat{\mathbf{x}}$ can be computed from the local element values provided the mesh element containing that point is known. The difficulty is thus finding the "basis element" of an arbitrary point $\hat{\mathbf{x}}$. Here a searching algorithm, based on area coordinates, is used to find the basis element. The algorithm is described [2], it proceeds by computing the barycentric coordinates of $\hat{\mathbf{x}}$ relative to the current element. If the point is not inside the element, then the direction in which to search is determined from the barycentric coordinates. The adjacent element in the given direction is thus searched next until the element containing the interpolation point is found. The searching procedure will typically start with a guessed element and traverse a band of elements as illustrated in Fig. 4.6. Once the basis element, the element shape functions presented in section 4.1.4 are used to compute $\psi(\hat{\mathbf{x}})$ by standard FEM interpolation.

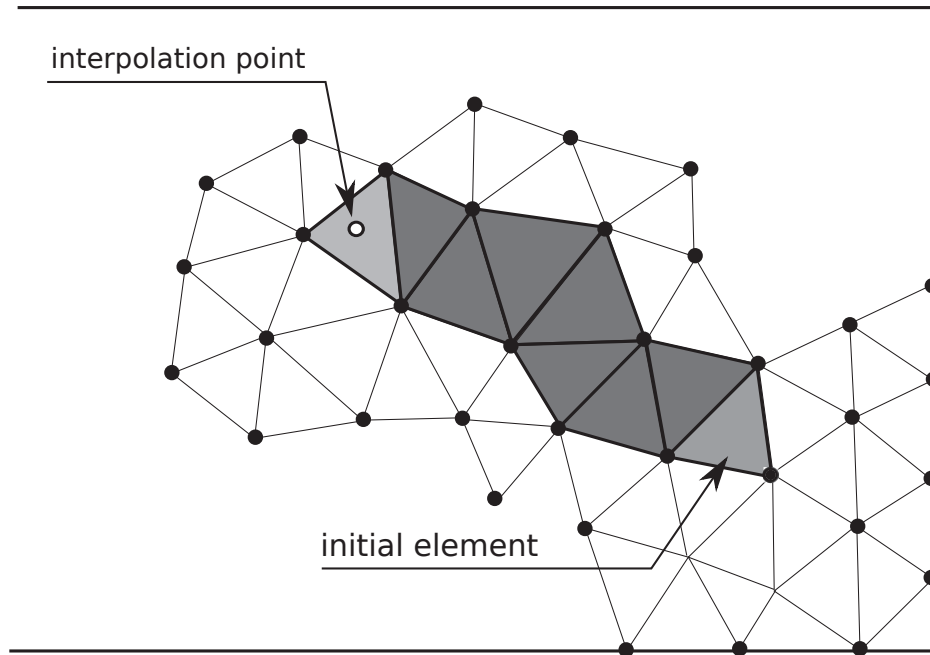


Figure 4.6: Element searching procedure.

4.11 Solution Algorithm

In the present section, the complete numerical algorithm is summarized. For all simulations performed in this thesis the following comments hold:

- a conventional two-time-levels finite differences procedure is used to integrate the equations through time.
- the discretized system (4.23) is solved in a coupled (monolithic) fashion for velocity and pressure.
- when not else specified the mini element ($P_1^{bubble} P_1$) was used.
- the energy equation is coupled one way to the Navier-Stokes equation in order to avoid the need to resort to an iterative procedure.
- the temperature is interpolated by linear shape functions with the same mesh nodes as the pressure.

Two different global solution approaches were used: (a) the governing equations are solved using the classic ALE approach, where the mesh velocity $\hat{\mathbf{v}}$ enters the convective term via Eq. (3.29), (b) continuous remeshing and interpolation are used to automatically account for the interface motion, without resorting to the ALE method. Most test cases presented hereafter have been simulated using both approaches and no significant difference could be observed hence serving as a cross validation of both methods. However, approach (b) appeared to be more robust because it applies remeshing at every time iteration immediately before the equations are solved thereby reducing the risk of mesh tangling. In the following the two approaches are going to be described.

Consider the vectors \mathbf{U}_h^n , \mathbf{T}_h^n and \mathbf{P}_h^n containing all the nodal values at time level n of the velocity, temperature and pressure respectively. The subscript h refers to a specific triangulation of the domain, which is defined by the vector \mathbf{X}_h^n containing all the mesh point's coordinates and the connectivity relating the mesh points to each mesh element. In the approach (a), the governing equations are solved at each time step just after the mesh has been moved, with the previous time step's solution appearing on the right hand side. Because of the ALE formulation, the previous time step's solution does not require interpolation. This is due to the fact that the local time derivative in (3.29) is defined for a fixed point in the moving mesh frame.

For the ALE approach (a), the following steps are performed at every time level n for a given solution $(\mathbf{U}_h^n, \mathbf{T}_h^n, \mathbf{P}_h^n)$ on the mesh h with points \mathbf{X}_h^n :

1. Compute the difference in heat flux on both sides of the interface and the corresponding \dot{m}'' based on \mathbf{T}_h^n
2. Compute mesh velocities \mathbf{U}_h^{mesh} as a function of $(\mathbf{U}_h^n, \mathbf{X}_h^n)$ and \dot{m}''
3. Calculate the time step Δt
4. Move every mesh point to its new position: $\mathbf{X}_h^{n+1} = \mathbf{X}_h^n + \Delta t \mathbf{U}_h^{mesh}$

5. Calculate the curvature, assemble matrices and vectors
6. Advance the temperature in time by solving the energy equation for T_h^{n+1}
7. Advance the flow quantities in time by solving the Navier-Stokes equations for U_h^{n+1}, P_h^{n+1}
8. Remesh (if needed), i.e. insert/delete mesh points and triangulate the new set of points, defining the mesh \tilde{h}
9. If remeshing was used: interpolate the solution on the new mesh: $(U_h^{n+1}, T_h^{n+1}, P_h^{n+1}) \rightarrow (U_{\tilde{h}}^{n+1}, T_{\tilde{h}}^{n+1}, P_{\tilde{h}}^{n+1})$

In the second approach (b), the solution of the flow equations is uncoupled from the mesh motion. This is possible since an interpolation is always applied after the mesh points have been moved. Therefore, the equations can be solved in their Eulerian form, i.e. with $\hat{\mathbf{v}} = \mathbf{0}$ in Eq. (3.29).

Approach (b) proceeds in a similar way and is thus only described for adiabatic simulations, where the following solution procedure is used:

Starting from a given solution (U_h^n, P_h^n) at time level n on the mesh h with points X_h^n

1. Compute mesh velocities U_h^{mesh} as a function of (U_h^n, X_h^n)
2. Calculate time step Δt
3. Move the mesh points to their new positions, according to: $X_h^{n+1} = X_h^n + \Delta t U_h^{mesh}$
4. Perform mesh operations (point insertion/deletion)
5. Remesh i.e. define connectivity of the new points: $X_{\tilde{h}}^{n+1}$
6. Interpolate old velocity on new mesh $U_h^n \rightarrow U_{\tilde{h}}^n$
7. Calculate curvature and assemble system arrays using $X_{\tilde{h}}^{n+1}$
8. Solve the Navier-Stokes equations on \tilde{h} to obtain the flow variables at the new time level $(U_{\tilde{h}}^{n+1}, P_{\tilde{h}}^{n+1})$

In approach (b) remeshing is applied at every time level and the solution is interpolated before the Eulerian equations are solved to advance the solution in time. Since regenerating the mesh takes time and the required interpolation introduces errors, approach (b) can only be beneficial in cases with highly irregular motion, where frequent remeshing would be required anyway. However, due to the generation of new connectivity arrays after mesh motion, tangling can almost always be avoided. Moreover, the additional time required by both the remeshing procedure and the successive interpolation was found to make up only a minor fraction of the total computational time, which is dominated by the solution of the linear system.

5 Computational Studies

This section presents many simulation results obtained with the method described above. Additional benchmarks, which are not displayed here, have been used to verify the spatial and temporal rates of convergence. The start-up of a Poiseuille flow showed that the results converged (with the expected rate) towards the analytical solution. Parts of this chapter have already been published in the article [56]. Before moving to two-phase flow, two single-phase problems are simulated to show the versatility and increase the confidence in the present approach.

5.1 Flow around a Sphere

In this section, the flow around a sphere, which moves at a constant velocity relative to an unbounded fluid, is studied. This is a classical problem in fluid dynamics for which an analytical solution is available in the Stokes flow limit, as well a large amount of reference simulations. The limit of vanishing Re , which possesses an analytical solution, is considered first.

5.1.1 Stokes Flow around a Sphere

Stokes flow describes the case where viscous forces, which have a characteristic size $\mu U/L$, are much larger than inertial forces of characteristic size ρU^2 . Formally, this is obtained by letting $Re \rightarrow 0$ in the Navier-Stokes equations Eq. (3.5)-(3.6), without body and surface tension forces. An analytical solution for the Stokes flow around a sphere was found by G. G. Stokes himself and published in his 1851 paper [129]. The analytical solution is given in spherical coordinates. The radial (u_r) and the tangential velocity (u_θ) components are given by

$$u_\theta = -U \sin(\theta) \left(1 - \frac{a^3}{4r^3} - \frac{3a}{4r} \right), \quad u_r = U \cos(\theta) \left(1 - \frac{3a}{2r} + \frac{a^3}{2r^3} \right), \quad (5.1)$$

where a is the sphere radius. The drag coefficient is defined as

$$C_D = \frac{F_D}{\pi a^2 \rho U^2 / 2}, \quad (5.2)$$

where F_D is the total drag force exerted on the sphere and using the solution (5.1) yields $C_D = 24/Re$.

The problem was simulated by numerically solving the axisymmetric Navier-Stokes equations with a small value of the Reynolds number: $Re = 2 \cdot 10^{-5}$. Figure 5.1 displays the computational domain and boundary condition. In order to avoid the influence of the artificial boundaries

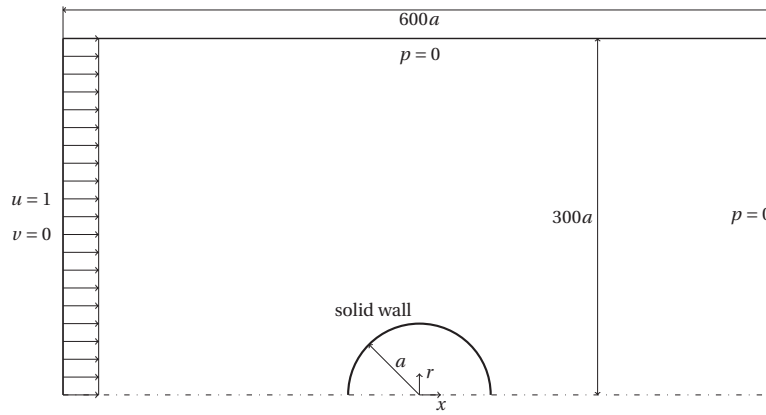


Figure 5.1: Computational domain and boundary conditions for the Stokes flow around a sphere.

on the flow close to the sphere, a large computational domain was used. The dimensions are $600a \times 300a$ in the axial and radial directions, respectively. The following boundary conditions were used: no-slip on the surface of the sphere, a constant axial velocity on the left boundary, symmetry on the bottom and top boundaries, and a constant pressure at the outlet. A view of the finite element mesh, which is gradually refined around the sphere, is displayed in Fig. 5.2. Figure 5.3 compares the computed steady state solution to the analytical expressions (5.1). The computed drag coefficient was $C_D = 1.164 \cdot 10^6$, which is close to the theoretical value obtained from Eq. (5.2) of $C_D = 1.2 \cdot 10^6$.

5.1.2 Flow around Sphere for $Re = 200$

The flow around a sphere was also simulated for a higher Reynolds number. Here, the non-linear convective term is not negligible and no analytical solution is known. However, experimental and numerical reference data is available. Tabata and Itakura [131] used a finite element method to compute drag coefficients in the range $10 < Re < 200$. Experimental measurements are available from Roos and Willmarth [112] as well as in some classic textbooks like Schlichting [117]. Depending on the Reynolds number different flow patterns can be observed. Below $Re = 20$ the flow is laminar and no separation occurs. For $20 < Re < 200$

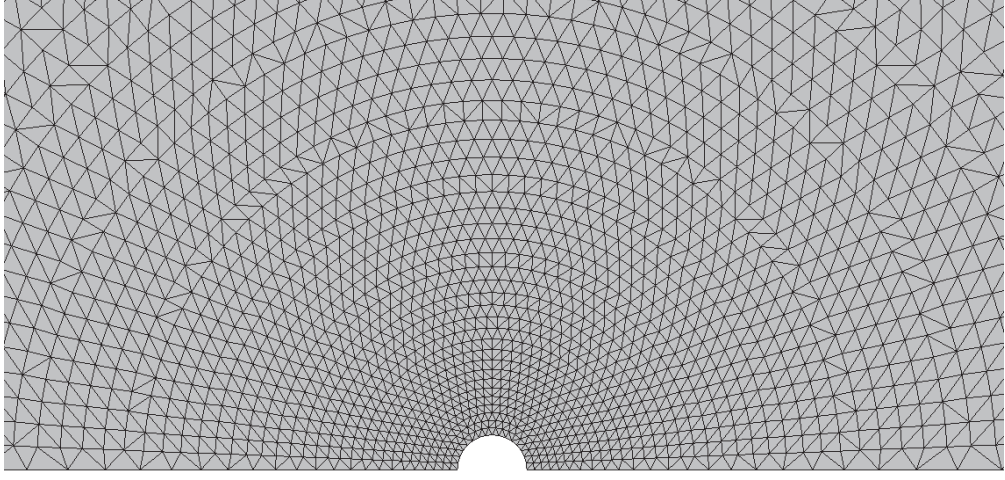


Figure 5.2: Finite element mesh for the Stokes flow around a sphere.

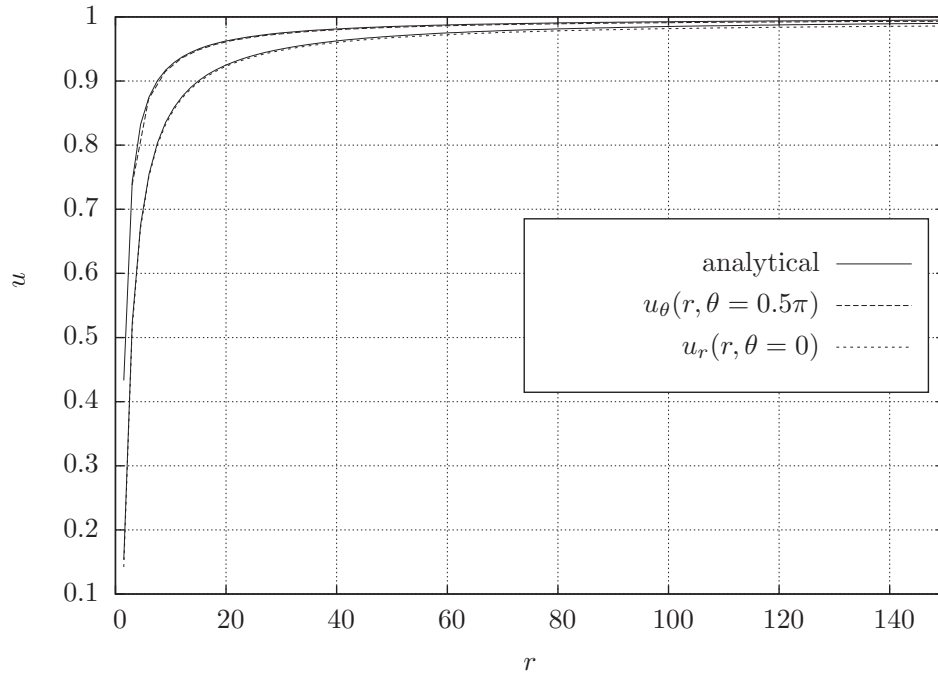


Figure 5.3: Stokes flow around a sphere: analytical and numerical solutions.

the flow separates and a region of closed streamlines appears behind the sphere but the flow remains steady and axisymmetric. Above $Re = 200$ the flow stops being axisymmetric [131] and for $Re > 270$ it stops being steady. Finally, for $Re > 1000$ the wake becomes turbulent.

Here we simulate the flow around the sphere for $Re = 200$, in the steady axisymmetric regime. Figure 5.4 shows the simulated flow field at steady state. The shapes of the streamlines in Fig. 5.4 is in qualitative agreement with those found in [5]. The drag coefficient was com-

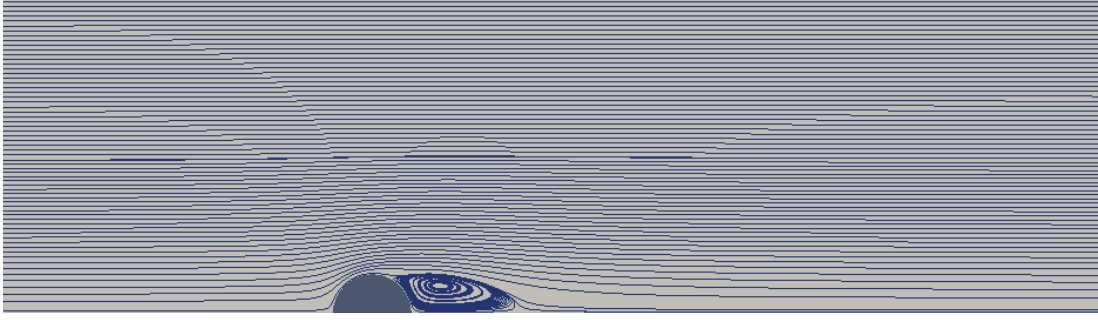


Figure 5.4: Streamlines for a flow around a sphere at $Re = 200$.

puted to be $C_D = 0.778$, which is close to the value obtained by [131] of $C_D \approx 0.77$ and the experimental value of [112, 117] of $C_D \approx 0.8$.

5.2 Natural Convection in a Heated Cavity

In order to verify the coupling between the Navier-Stokes and energy equations, a single-phase natural convection problem is simulated. The left and right walls of a two-dimensional cavity have a fixed temperature of $T_l = -0.5$ and $T_r = 0.5$ respectively. The top and bottom walls are assumed to be adiabatic i.e. $\dot{q}'' = -\lambda \partial_n T = 0$. Figure 5.5 shows a sketch of the problem. Zero flow and zero temperature are used as initial condition. At $t = 0$ the fluid begins to rise

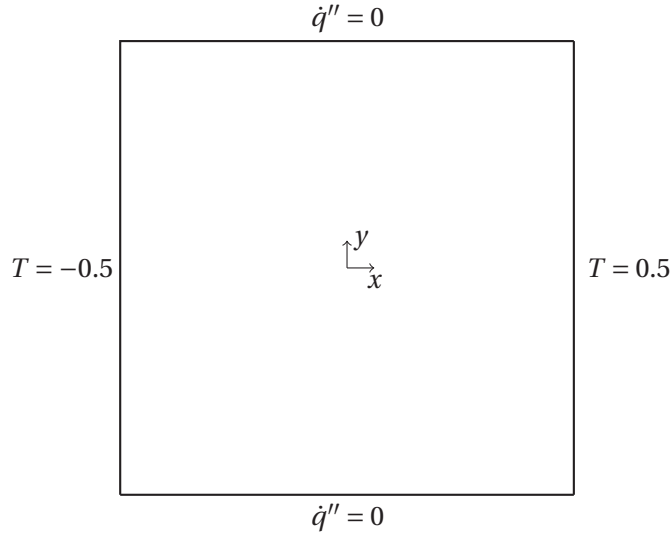


Figure 5.5: Schematic of the cavity benchmark problem.

on the right because it is heated and to fall on the left where it is cooled, thus a circulation is started. The temperature dependence of density is included in the momentum equation with

the Boussinesq approximation:

$$\rho g = \rho_0(1 - \beta T), \quad (5.3)$$

where ρ_0 is the reference density and β is a constant.

The Prandtl and Rayleigh numbers are:

$$Pr = \frac{\mu c_p}{\lambda} = 0.71, \quad Ra = \frac{\rho g \beta L^3}{\mu \alpha} \Delta T = 10000. \quad (5.4)$$

Figure 5.6 shows the streamlines and the temperature distribution when steady state is reached. The horizontal velocity on a vertical line is plotted in Fig 5.7, where it is compared to the simulation results of Corzo [28] displaying a very accurate match.

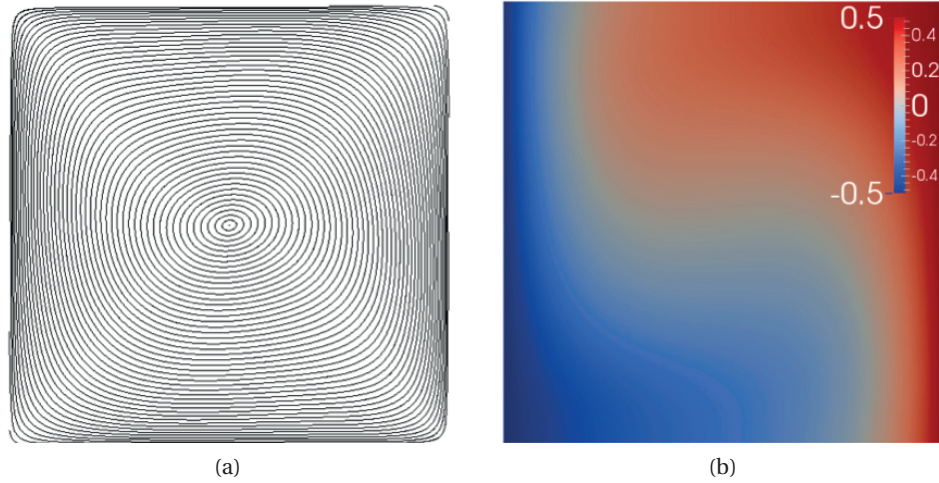


Figure 5.6: (a) Streamlines and (b) temperature distribution - inside the heated cavity.

5.3 Annular Poiseuille Flow

The Poiseuille flow is a classical benchmark case for single-phase numerical codes. Therefore, we simulate its two-phase counterpart that is a layered Poiseuille flow in a cylindrical pipe. The setup is described in Fig. 5.8 and the following assumptions hold:

- steady, fully developed conditions ($\partial_t = \partial_x = 0$).
- no gravity, constant pressure gradient: $\partial_x p = -F$.
- the two layers are concentric leading to axial symmetry around $r = 0$.
- piecewise constant material properties in the core $0 < r < R_i$ (region 1) and the film $R_i < r < R$ (region 2), where R is the radius of the pipe and r is the radial coordinate.

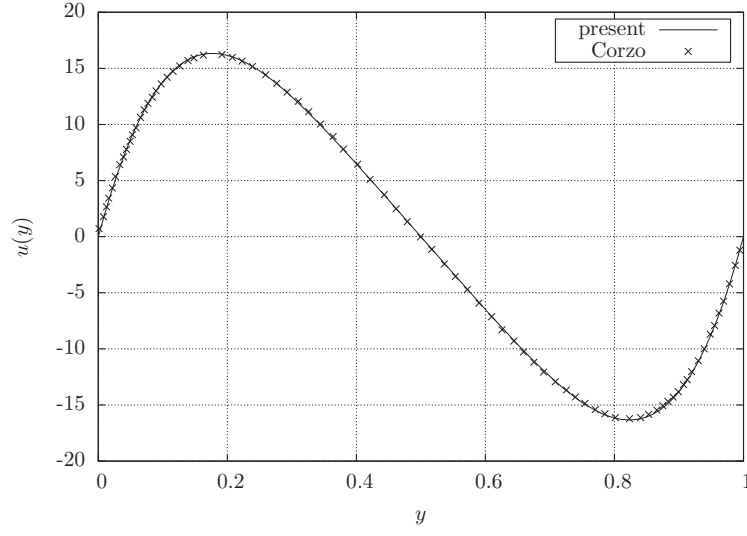


Figure 5.7: Horizontal velocity along the line $x = 0$.

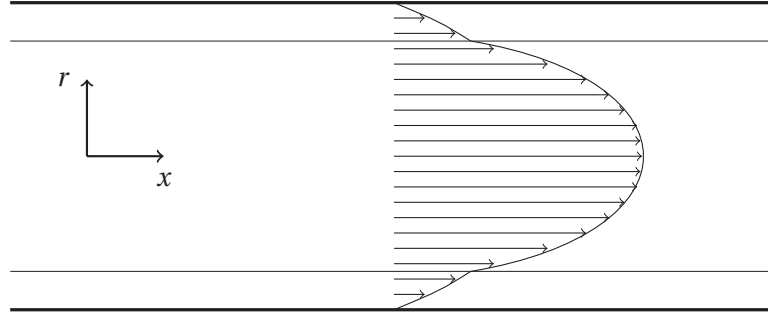


Figure 5.8: Schematic of the annular Poiseuille flow.

An analytical solution is easily derived from the axisymmetric momentum equation, leading to the (non-dimensional) velocity profile:

$$\begin{aligned}
 u(r) &= u_0 \left(1 - \frac{m(r/R_i)^2}{a^2 + m - 1} \right) \quad \text{for } 0 < r < R_i, \\
 u(r) &= u_0 \frac{a^2 - (r/R_i)^2}{a^2 + m - 1} \quad \text{for } R_i < r < R,
 \end{aligned} \tag{5.5}$$

where $m = \mu_2/\mu_1$ is the viscosity ratio and $a = R/R_i$. Moreover, from the balance of forces in the axial direction, the (non-dimensional) pressure gradient follows as:

$$F = \frac{4 u_0}{Re(a^2 + m - 1)R_i^2}. \tag{5.6}$$

The axisymmetric Poiseuille flow was simulated by applying a constant pressure gradient of $F = \frac{32}{Re}$ over the channel length. This was realized by imposing the pressure as a boundary condition in the inlet and outlet sections. The geometry is non-dimensionalized based on the diameter of the channel such that $R = 0.5$ and the location of the interface is chosen as $R_i = 0.35$. The Reynolds number for the simulations was set to 20 and the viscosity ratio is $m = 100$. The numerical results are compared to the steady analytical solution after the transient have decayed.

The analytical solution (5.5) is a piecewise quadratic parabola, which should be interpolated exactly by the quadratic element shape functions. This was verified by simulating the problem with the quadratic finite element described in section 4.1.4. Figure 5.9 compares the velocity profiles obtained from simulations using the mini element and the quadratic element to the analytical solution (5.5). Despite both simulations were performed with the same mesh, the velocity profile obtained with the mini element is accurate but less accurate than the profile obtained with the quadratic element. The error with the quadratic element is indeed on the order of the machine precision.

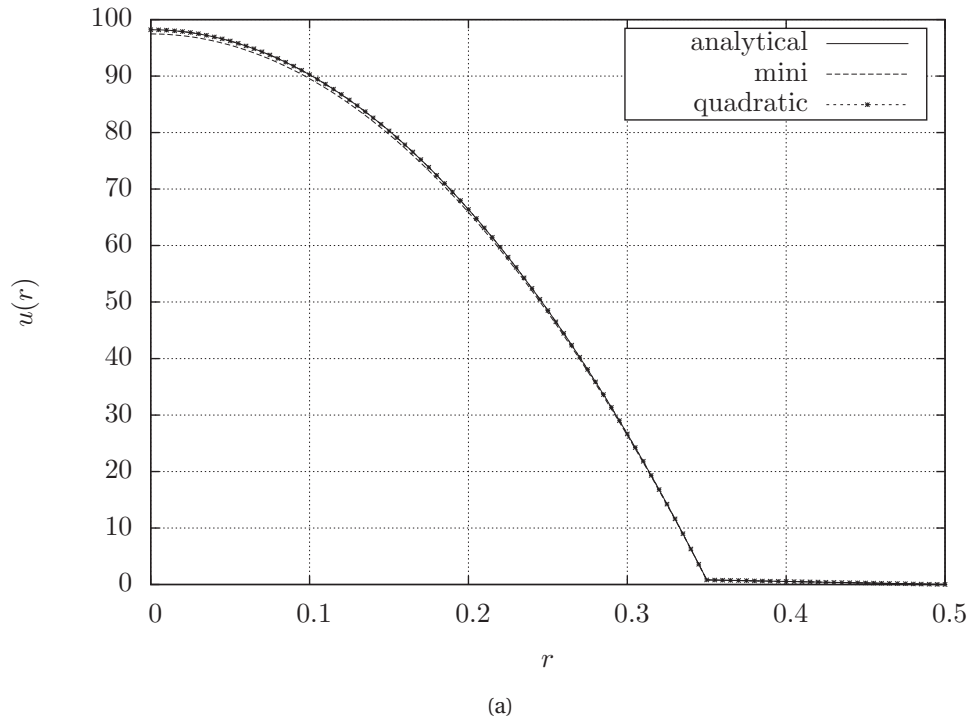


Figure 5.9: Axial velocity profile of the annular Poiseuille flow: analytical solution Vs two finite element discretizations (mini element and quadratic element).

5.4 Static Drop without Spurious Currents

The static drop benchmark is a standard benchmark in numerical two-phase flow. In fact most methods used to capture or track the interface fail to provide a net balance between pressure and surface tension forces even when the velocity field is zero. This results in a local non-zero acceleration of the flow at the interface and thus, in the creation of spurious currents. These currents, which are sometimes referred to as parasitic currents, can generate unphysical movement of the interface. It is therefore the aim of any numerical method to minimize spurious currents.

The static drop is immersed in a constant background velocity in order to verify the interplay between interface advection and surface tension calculation as suggested in [99]. There are no external forces and no pressure gradient and thus the flow is purely driven by inertia. In the absence of gravity and velocity gradients, the only non-vanishing terms in the momentum equation are the pressure and surface tension. The Navier-Stokes equations thus reduce to the Young-Laplace equation, which for a spherical drop reads:

$$\Delta p = \frac{\kappa}{We} = \frac{4}{WeD}. \quad (5.7)$$

The Young-Laplace equation (5.7) is given in non-dimensional form, where D is the non-dimensional diameter of the drop and Δp is the non-dimensional pressure difference across the interface. The accuracy of the surface tension calculation is assessed by evaluating the interfacial pressure jump and the magnitude of the spurious currents.

The computational domain is chosen with a size of 10×1 in the axial and radial directions respectively, with the diameter of the drop $D = 1$. Symmetry boundary conditions are used on the symmetry axis ($r = 0$) and on the upper boundary, while on the left and right boundaries Dirichlet boundary conditions are used setting the velocity to $\mathbf{v} = (1, 0)^T$. Additionally, in order to get a unique solution, the pressure value is fixed at the upper right corner of the domain. The drop is initialized as a sphere with center located one diameter from the left boundary and the simulation is run until the front of the drop reaches the right boundary of the domain. The axisymmetric Navier-Stokes equations are solved for two phases with large density and viscosity ratios of 0.001 and 0.01, respectively, as these values are similar to those for a water drop in air. The non-dimensional numbers Re , We are set to one.

Figure 5.10 displays the translating drop at different instants of time on a mesh discretizing the interface with 16 uniform line segments. The pressure distribution for the middle frame, i.e. $t = 3.55$, is visualized in Fig. 5.11. The full flow velocity was used to move the interface with $\gamma = 0$ in Eq. (4.28). We consider the velocity $e_{\mathbf{v}}$ and pressure $e_{\Delta p}$ errors defined as follows

$$e_{\mathbf{v}} = \max_T \left(\max_{\Omega} (||\mathbf{v}_h - \mathbf{v}_{\infty}||) \right), \quad e_{\Delta p} = \max_T \left(\frac{|\Delta \bar{p}_h - \frac{4}{WeD}|}{\frac{4}{WeD}} \right), \quad (5.8)$$

where the pressure jump $\Delta \bar{p}_h = \bar{p}_h^i - \bar{p}_h^o$ is computed as the difference between the average

pressures inside and outside the bubble. The maxima in (5.8) are taken over the nodes of the computational domain Ω and over the collection of computed time steps T . Table 5.1 displays the errors for different mesh resolutions, where the half circle representing the drop is discretized with line segments of uniform length and n denotes the number of segments. The spurious currents and the errors in the pressure jump displayed in Table 5.1 are only on the order of the computer's arithmetic precision independently of the mesh resolution. The vanishing of the error measures $e_{\mathbf{v}}$ and $e_{\Delta p}$ is possible due to a consistent discretization of the pressure and surface tension term. This is enabled by using the same FEM basis functions for the pressure and the heaviside function [27]. The pressure gradient can thus be computed in the same way as the surface tension term, which is proportional to the gradient of the heaviside function H_h , and $\nabla p_h = \kappa \nabla H_h$ can be satisfied numerically. Provided the exact curvature κ is available the value of the pressure jump can then be computed exactly. The

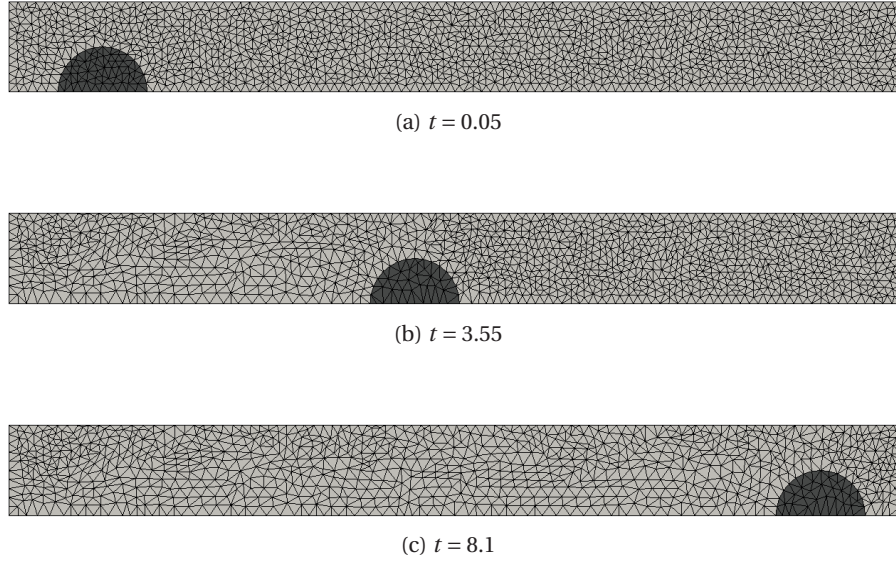


Figure 5.10: Translating drop and mesh at different instants.

initial drop shape is represented by a half circle with uniformly spaced mesh points on its circumference, which is a particular case where the curvature formulas Eq. (4.13) and (4.14) are exact. It has been verified numerically that the curvature deviated from the exact curvature of the circle, which is $2/D$, only by the order of the machine accuracy. Moreover, the motion of the interface and the remeshing do not introduce any error in the computed interface position. The former is due to the fact that the full fluid velocity is used to move the interface mesh and the latter is because when points are inserted on the interface mesh they are inserted on a circular arc taking the local curvature into account. Therefore, the computed curvature is always exact and the drop can reach the right end of the domain while essentially recovering the analytical solution at every time step. Removing the tangential velocity fully or partly, i.e. $\gamma \neq 0$ in Eq. (4.28), was found to introduce an error in the translated interface location. This error led to the calculation of an inexact interface curvature and therefore to an error in the

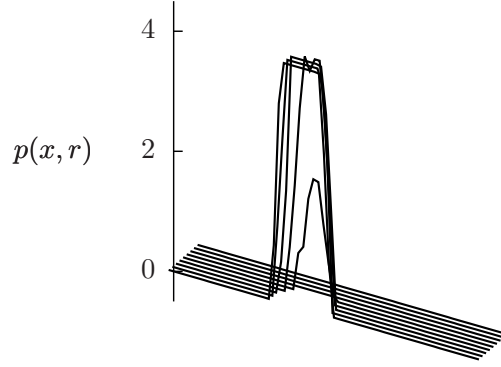


Figure 5.11: Pressure distribution at time $t = 3.55$.

pressure solution.

n :	2	4	8	16
e_{ν} :	10^{-14}	10^{-12}	10^{-13}	10^{-13}
$e_{\Delta p}$:	10^{-15}	10^{-13}	10^{-14}	10^{-14}

Table 5.1: Order of magnitude of the spurious currents and pressure errors.

5.5 Oscillating Droplet

In this section, we simulate the motion of a liquid droplet, which oscillates freely around its spherical equilibrium shape in the absence of gravity. When the drop is released from rest with an initial perturbation capillary forces lead to oscillations on its surface. Lamb [81] derived the frequency of small amplitude oscillations of a liquid globe by assuming perturbations of the form: $Y_n^m(\theta, \phi) \cos(\omega t)$, where $Y_n^m(\theta, \phi)$ are spherical harmonics. Here we restrict our attention to axisymmetric perturbations, wherefore $m = 0$. The inviscid dispersion relation for a droplet immersed in an outer fluid is given in §275 [81] and reads as:

$$\omega^2 = \frac{1}{WeR^3} \frac{n(n+1)(n-1)(n+2)}{(n+1)\rho_{in} + n\rho_{out}},$$

where R is the non-dimensional equilibrium radius of the drop. The viscous case was also investigated by Lamb in §355 [81] but without outer fluid and he found that free surface oscillations of a liquid drop are damped exponentially with a damping time of

$$\tau = \frac{ReR^2}{(n-1)(2n+1)}. \quad (5.9)$$

The least damped mode ($n = 2$) is considered here, for which

$$\omega^2 = \frac{24}{WeR^3(3\rho_{in} + 2\rho_{out})}, \quad \tau = \frac{ReR^2}{5}. \quad (5.10)$$

At the beginning of the computation the drop is a spheroid with half-axis $a = R + \hat{a}_0$ and $b = R - \hat{b}_0$, which are chosen such that the volume is equal to that of a sphere with radius R . The density and viscosity of the inner fluid are chosen 1000 larger than those of the outer fluid, such that the outer phase is negligible, which is required for Eq. (5.9) to hold. Small initial perturbations \hat{a}_0, \hat{b}_0 are applied in order to compare the simulations with the results from linear analysis. Varying the size of the initial perturbations showed that $\hat{a}_0 \approx 0.0034 R$ and $\hat{b}_0 \approx 0.0066 R$ was sufficiently small. The simulation results are compared to a damped oscillation modes of the form:

$$a(t) = R + \hat{a}_0 \cos(\omega t) e^{-\frac{t}{\tau}}, \quad b(t) = R - \hat{b}_0 \cos(\omega t) e^{-\frac{t}{\tau}}, \quad (5.11)$$

where $a(t), b(t)$ are the time dependent lengths of the two half-axis of the drop. The equilibrium radius is computed from volume conservation as $R = (\frac{3V_0}{4\pi})^{1/3}$, where $V_0 = (4/3)\pi a_0^2 b_0$ is the initial volume. Note that the drop cannot remain spheroidal, as the volume of a spheroid with half-axis given by Eq. (5.11) would deviate from the initial volume V_0 . However, we are interested in the frequency ω and damping factor τ of the oscillation and not in the shape of the droplet. The Weber number is set to one and the Reynolds number is 100 corresponding to the regime where the drop is not damped aperiodically. A mesh with edge length $0.025R$ on the interface is used, as this was found to be fine enough to obtain converged results. The computed oscillations around the equilibrium position are compared to the ansatz (5.11) in Fig. 5.12, where the monitored interface positions are the half axis of the drop. We observe close agreement for the frequency but the simulation results seem to be less damped than predicted by Eq. (5.9). This is rather surprising given that numerical dissipation should be introduced by the numerical discretization. However, the discrepancy in the damping rate is rather small and could be due to initial effects. In fact, for ease of setting up the computations, the drop was initialized as an ellipsoid in a quiescent velocity field which differs from the assumptions made in [81]. Moreover, free oscillations of a liquid drop have time dependent values of the frequency $\omega(t)$ and damping parameter $\tau(t)$ [103], which reach the values derived by normal-mode analysis only for large times.

5.6 Rising Bubble

The 2D rising bubble benchmark of Hysing et Al [70] has become a standard benchmark in the numerical two-phase flow community. In the original paper, different simulation codes (from 3 different groups) were compared for rising bubble simulations. Here we reproduce case 1 of Table 1 in [70]. A bubble of fluid that is less dense than the surrounding liquid starts to rise due to buoyancy. The initial bubble is spherical and at rest in the computational domain shown in Fig. 5.13, whose dimensions are $L = 4$, $R = 1$ and the bubble has a unit diameter. Since the

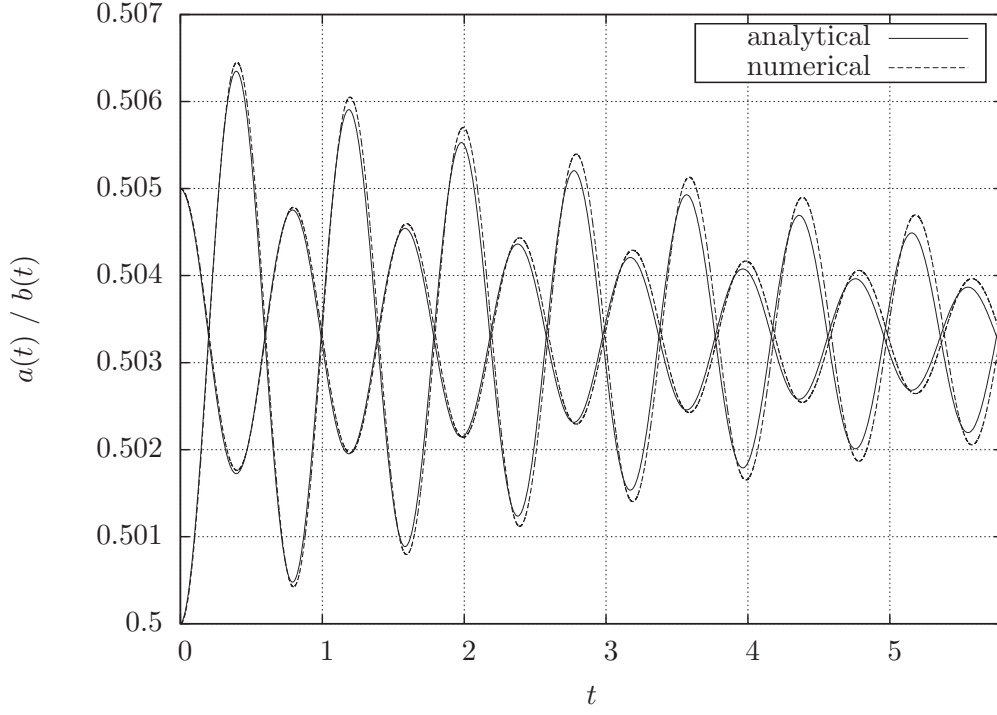


Figure 5.12: Lengths of the oscillating drop's half-axis (a and b) as a function of time.

geometry is symmetric and the flow was found to remain symmetric, we simulate the upper half of the problem. The boundary conditions are: no-slip on the top and bottom boundaries and slip on the side boundary and symmetry axis. The non-dimensional parameters for this case are: $Re = 35$, $We = 10$ and the density and viscosity ratio are both 10. Four different computational meshes are used to study the convergence behaviour.

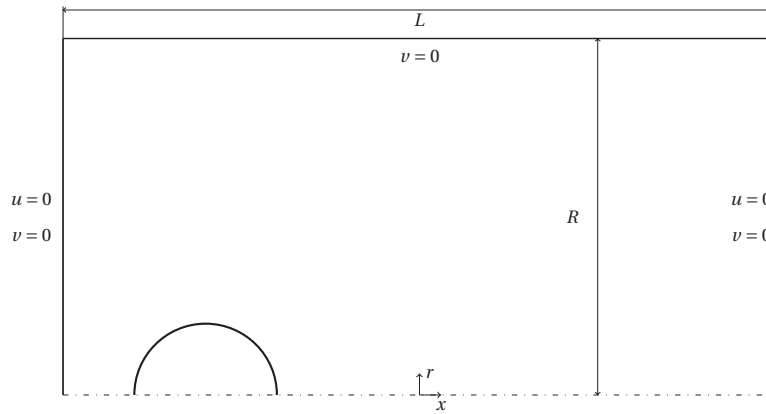


Figure 5.13: Schematic of the initial rising bubble geometry and boundary conditions.

As validation quantities, the rising velocity shape of the bubble are considered. Since we use unstructured meshes with remeshing, the number of mesh degrees of freedom can vary

strongly from one time step to another. Therefore, similarly to Group 3 in [70] we use the number of mesh points resolving the interface at the beginning $NDOF_{int}$ of the simulation as a measure for the mesh refinement level. Four meshes are used with $NDOF_{int} = 54, 79, 158, 316$ mesh edges on the initial interface, which because of our domain is only half the circle. In all cases a fixed time step size of $\Delta t = 0.0005$ was used. The rising bubble's centroid velocity is plotted in Fig. 5.14 and compared to the reference results of the simulation code MooNMD in [70]. All the groups in [70] obtained very similar results for the present test case. The same level of agreement is also observed with the present results. Note that our coarsest mesh, which has only 54 degrees of freedom on the interface, already gives very accurate results. Moreover, the results are observed to approach the reference shape as the mesh is refined. Figure 5.15 displays the shape of the bubble at time $t = 3$ with the different meshes. The present shape agrees well with that of MooNMD in [70] and the agreement gets better on finer meshes.

In this test case remeshing was used at every time iteration and the subsequent interpolation can introduce errors that lead to a significant volume conservation error if nothing is done to artificially correct it. This problem was absent in the previous test-cases, where the shape of the bubble was either an exact circle or no remeshing was needed. Table 5.2 displays the maximum relative deviation of the bubble volume $V(t)$, over the simulation time $t \in [0, 3]$, for the different meshes. Although the results presented in this section were obtained without any volume correction, the volume varied only very little from the beginning to the end of the simulation. The data in Tab. 5.2 shows that the relative variation of the bubble volume over the simulation time was below 0.8% with the coarsest mesh and below 0.3% with the finest mesh.

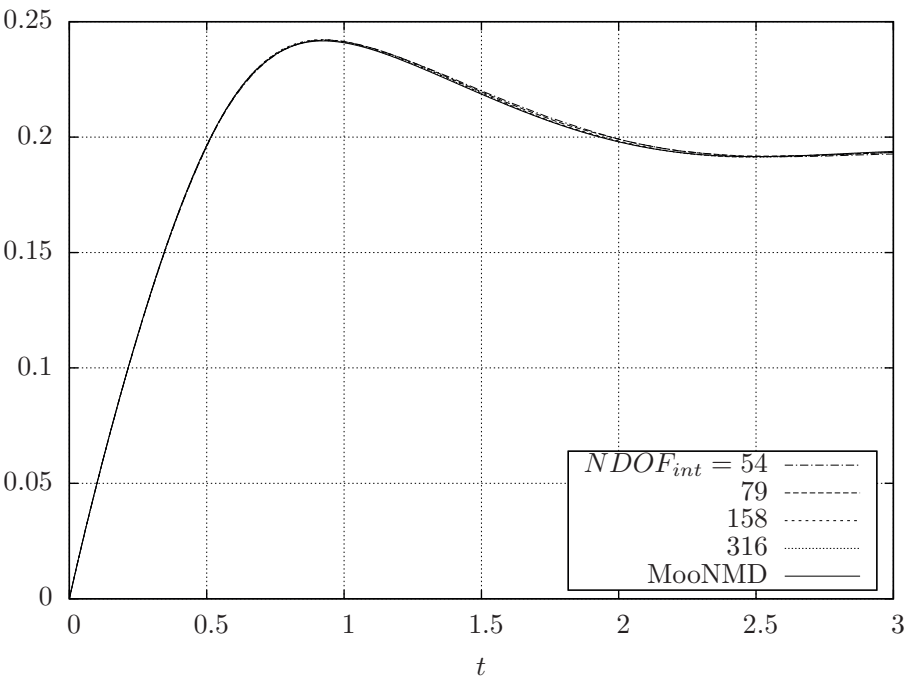
$NDOF_{int} :$	54	79	158	316
$\max_{t \in [0,3]} \left(\frac{V(t) - V_0}{V_0} \right) :$	$7.8 \cdot 10^{-3}$	$4.8 \cdot 10^{-3}$	$3.3 \cdot 10^{-3}$	$2.4 \cdot 10^{-3}$

Table 5.2: Volume errors with different levels of mesh refinement.

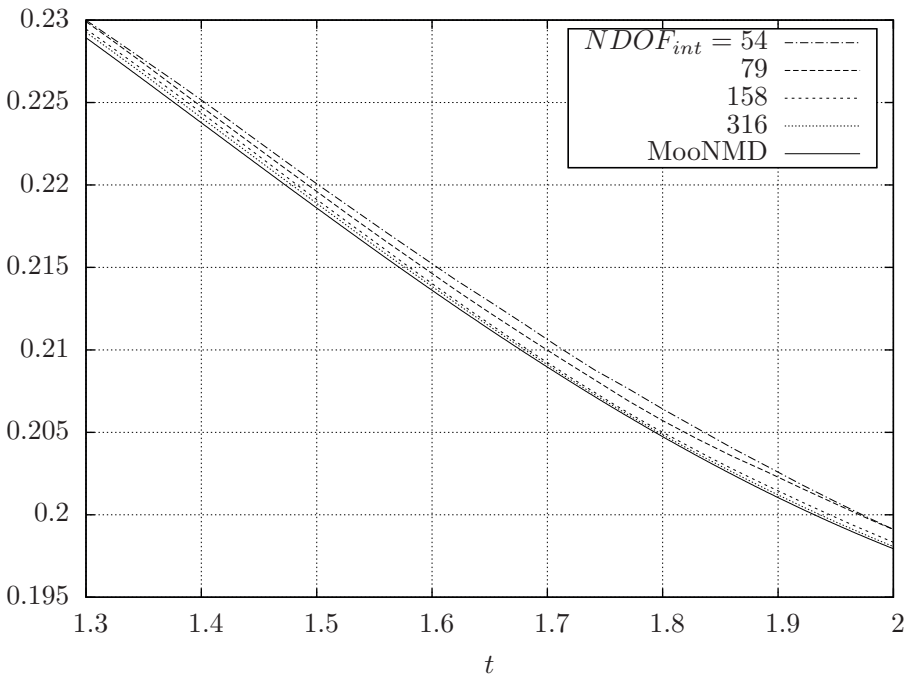
5.7 2D Rayleigh-Taylor Instability

In this section we investigate the unstable equilibrium of a heavy fluid is placed above a light fluid in a gravitational field. This situation is linearly unstable and leads to the famous Rayleigh-Taylor instability. The slightest perturbation from a perfectly flat interface will be amplified with an initial exponential growth. As the perturbation grows it will deviate from the exponential growth due to non-linear effects and the interface will form mushroom shaped fingers of the denser phase. The resulting complicated interface deformation is a good test case for the present moving mesh method.

We consider the two-dimensional case with no surface tension acting on the interface as in the conditions simulated by [84] using an interface capturing method and by [100] using a front

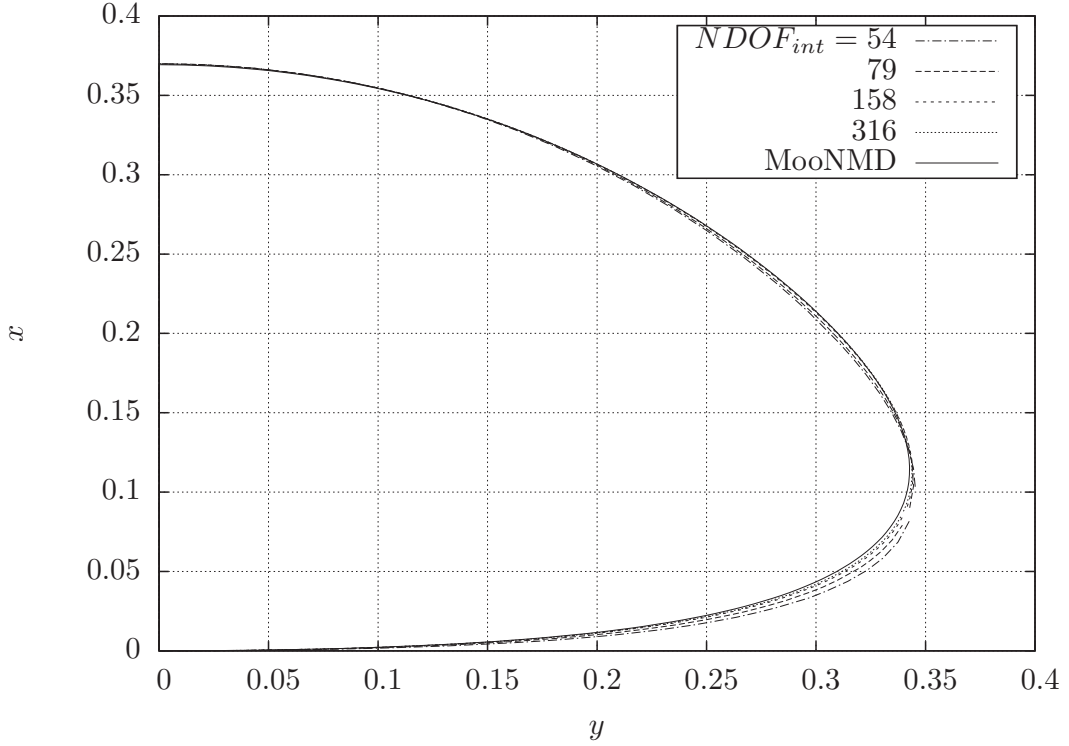


(a)



(b)

Figure 5.14: (a) Evolution of the rising bubble centroid velocity, for different mesh resolutions, (b) close-up view.


 Figure 5.15: Bubble shape at $t = 3$, for different mesh resolutions.

tracking algorithm. The density ratio of the fluids is 0.1383, their viscosities are equal and the non-dimensional numbers are $Re = 391.37$, $Fr = 0.31928$. A rectangular domain of dimensions 1×4 was used, where the interface was initialized with a sinusoidal perturbation of amplitude 0.05 and symmetry boundary conditions were used on all domain boundaries. The computed shape of the interface is displayed in Fig. 5.16 at different instants. The typical mushroom shape is observed at intermediate times and at later times, thin filaments are generated which are still resolved by the mesh. Qualitative agreement is found when comparing Fig. 5.16 with the figures 9 and 10 in [100]. In Fig. 5.17 the present shapes are compared to those found in figure 14 from reference [84]. The observed agreement demonstrates that the present method works also for cases with strong interface deformation.

5.8 Axisymmetric Rayleigh-Taylor Instability

In this test case the Rayleigh-Taylor instability is considered in an axisymmetric configuration where both fluids are contained in a cylindrical tube whose symmetry axis is oriented along the direction of gravity. Due to an initial perturbation the light fluid forms fingers, which are rising in the middle of the tube while the heavier fluid forms spikes falling along the tube wall. The exponential growth rate from linear stability analysis is observed at early times and at later times balance is established between buoyancy and form drag forces such that the rising

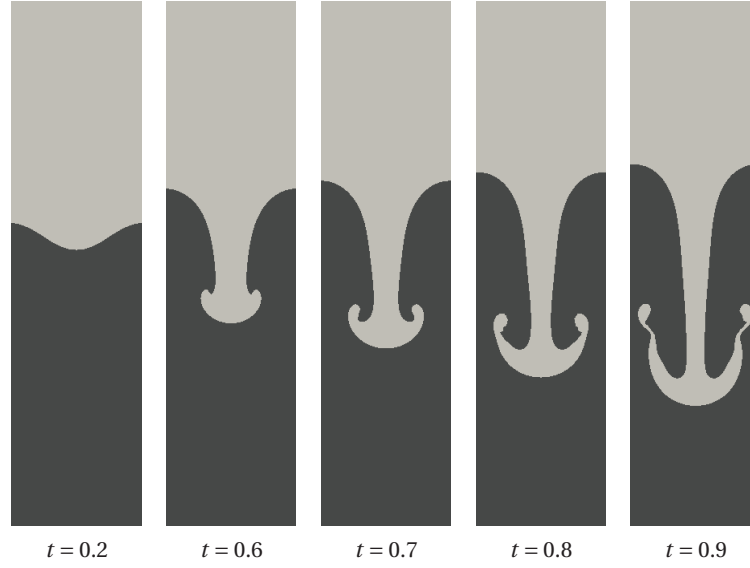


Figure 5.16: Interface shape at different instants for the Rayleigh-Taylor instability.

fingers reach a constant terminal velocity.

The bubble terminal velocity has been investigated by Davies and Taylor [32] who performed an experimental and theoretical investigation for a single axisymmetric air bubble rising in a cylindrical tube filled with water. As the tube was closed at the top and open at the bottom the bubble had no rear. Using inviscid irrotational flow Davis and Taylor derived an expression for the velocity of the interface vertex on the symmetry axis:

$$V_b = C\sqrt{g_0 R}, \quad (5.12)$$

where R is the tube radius and g_0 is the gravitational constant. Davies and Taylor [32] calculated a value of $C = 0.464$ and found good agreement with their experiments on emptying tubes, which yielded values of C ranging from 0.4 to 0.49. Another value of $C = 0.511$ was derived by Layzer [82] assuming incompressible irrotational flow. The two derived values differ because Davies and Taylor used Bernoulli's equation at an interface point with distance $R/2$ from the axis while Layzer required that Bernoulli's equation to hold in a first order neighbourhood of the bubble nose vertex. Both theoretical studies assumed a free boundary problem with a single inviscid incompressible fluid.

Here, we compare the initial growth rate of the perturbation and the final velocity of an interface vertex on the symmetry axis to the theoretical predictions. To simulate the problem, we use a domain of size $x \in [0, 4]$, $r \in [0, 0.5]$ with rotational symmetry around the x -axis and gravity pointing towards positive x . An initial perturbation of the interface is prescribed as:

$$x(t = 0, r) = 2.4 + A_0 J_0(\beta_1 r / R), \quad (5.13)$$

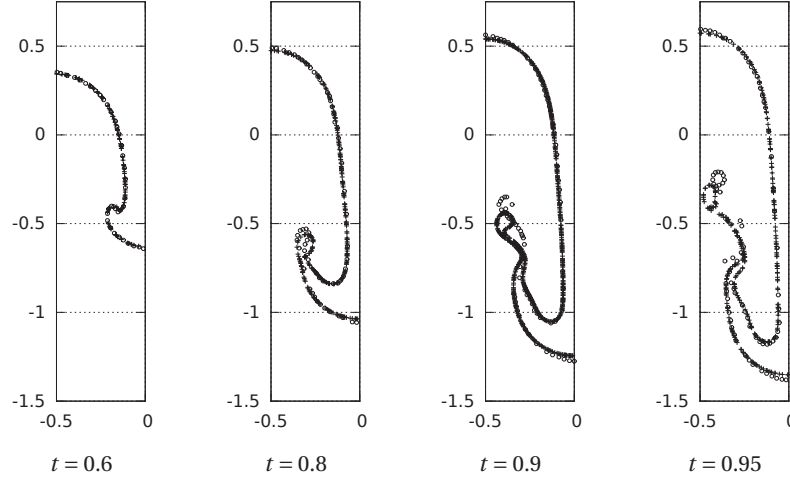


Figure 5.17: Shape comparison for the Rayleigh-Taylor instability, circles are the results from [84] and crosses are the present results.

where $J_m(r)$ is the Bessel function of first kind of order m and $\beta_1 \approx 3.8317$ is the smallest positive root of $J_1(r) = 0$. We set the density of the heavy phase to 1 and the light phase has zero density. Gravity is given by $Fr = 1$ and surface tension is turned off for this calculation. In order to have nearly inviscid flow we set $Re = 1000$. The time step is chosen as $\Delta t = 0.01$ and the mesh has a characteristic size of 0.005 on the interface. Fig. 5.18 displays the time evolution of the interface and the mesh. The position of the interface vertex on the symmetry axis $x(t, r = 0)$, for $A_0 = 0.0005$, is compared to Eq. (5.12) in Fig. 5.19. The exponential growth obtained from inviscid linear stability analysis: $x(t, r = 0) = \exp(t\sqrt{\beta_1 g/R})$ is also plotted in Fig. 5.19.

After an initial phase of accelerated growth the front reaches a regime, where the streamlines pattern is time independent and the rising velocity is constant. At early times, when the perturbations are still small and non-linear effects negligible, the numerical results are expected to follow the exponential growth from linear stability analysis. The agreement of initial growth rate with linear stability analysis was found to improve significantly by using an initial condition including the velocity perturbation described by the corresponding eigenfunctions of the linear stability problem. As can be seen in the inset of Fig. 5.19, the initial growth rate is slightly lower than the one from linear stability analysis probably due to viscous effects. Viscous effects will be less significant if the Reynolds number is increased. However, simulations with higher Reynolds number could not be carried out due to numerical instability reasons. The simulated terminal velocity in Fig. 5.19 follows Eq. (5.12) with a value of C right in between the values given by Taylor and Layzer.

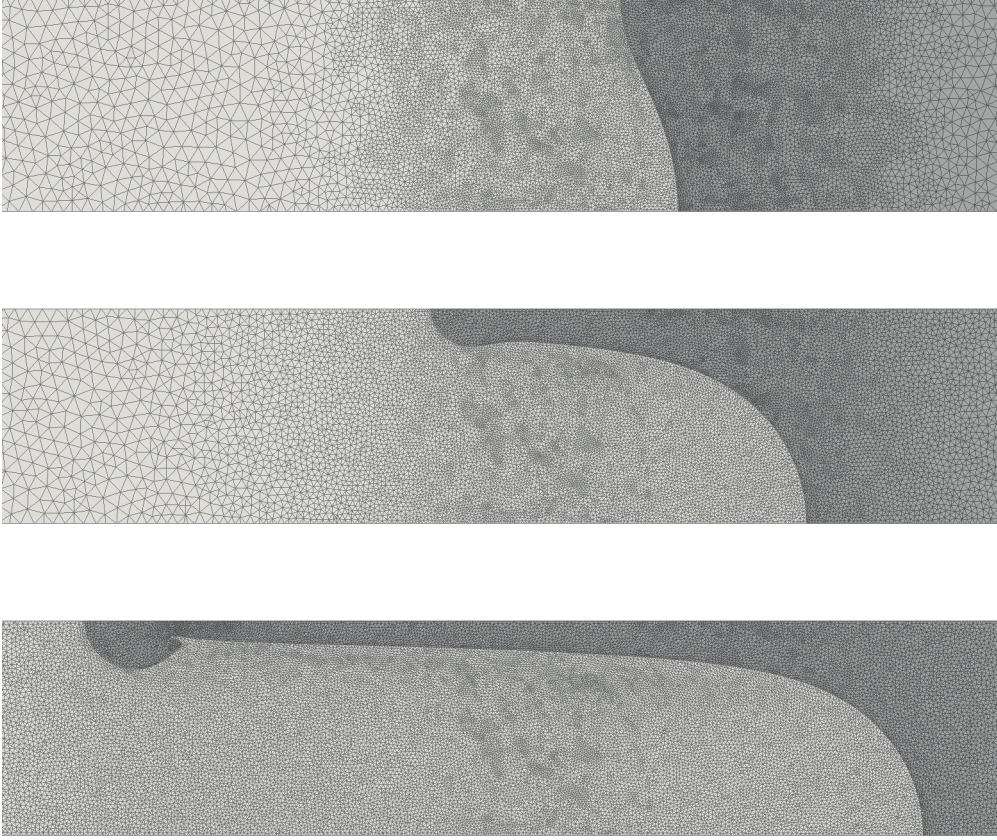


Figure 5.18: Evolution of the cylindrical Rayleigh-Taylor instability.

5.9 Travelling Waves in Liquid Film Falling on the Surface of a Fiber

In this section we consider a liquid film falling under the influence of gravity over the surface of a vertical cylindrical fiber. This is an unstable flow, which exhibits interesting dynamical features including travelling waves and solitary waves. The setup shown in Fig. 5.20 was studied in [38], where experimental results were compared to a weighted residual, long-wave-asymptotic model developed by the authors. They investigated the spatial response of this convectively unstable flow (noise amplifier) at moderate Reynolds number, with periodic inlet forcing and naturally excited films were also considered. Depending on the ratio of the forcing frequency f_{for} to the frequency of maximum linear growth rate f_M , two regimes were found. For $f_{for} \approx f_M$ the primary instability led directly to the formation of a saturated wave-train with frequency f_{for} , whereas for $f_{for} < f_M$ a sequence of coalescence events preceded the final wave regime.

The weighted residual model, used in [38], contains two equations corresponding to axial momentum balance and mass conservation and two independent variables, namely the film thickness and the flow rate. It is based on an weighted integration of the momentum equation,

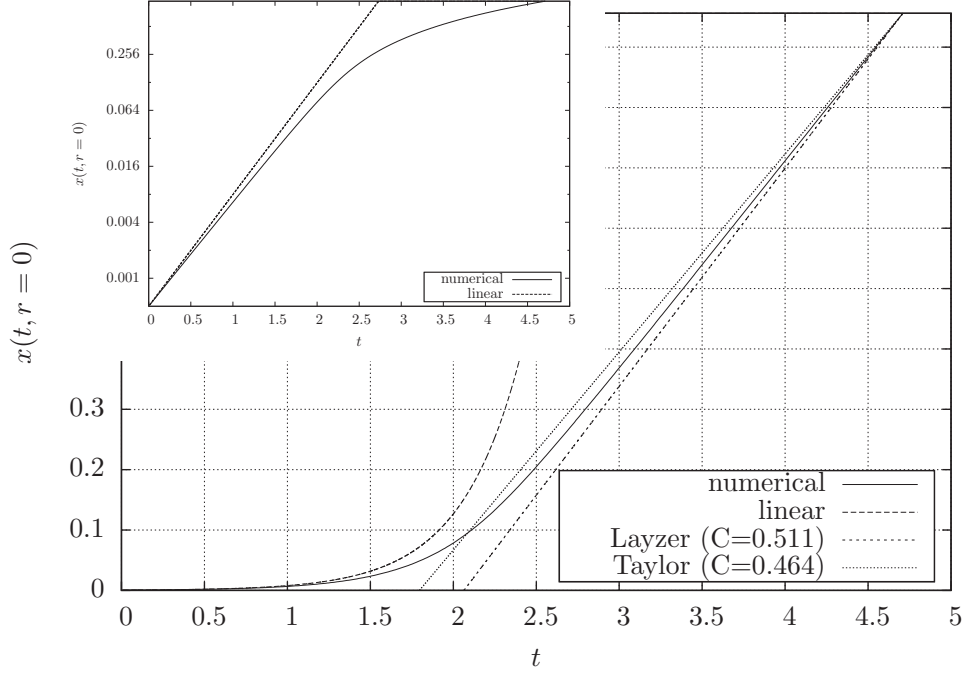


Figure 5.19: Position of the bubble nose as a function of time, the inset has a logarithmically scaled ordinate.

where the velocity profile is expanded on a set of polynomials, which are at the same time used as weight functions like in the Galerkin method. The time ∂_t and axial space ∂_x derivatives are assumed small by introducing a film parameter ϵ . The long wave approximation and the weighted residual integration result in a system of two coupled PDE for the film thickness h and the flow rate q , which is consistent up to ϵ^2 and includes inertia, wall friction, gravity, surface tension and viscous dispersion effects [38]. By assuming the solution to be invariant along constant velocity trajectories, the PDE system reduces to a system of ODE. Travelling wave solutions can then be computed as limit cycles (finite wavelength) and solitary waves as homoclinic orbits (infinite wavelength) of the corresponding dynamical system.

In this section we simulate the conditions of Fig 22 (a) in the paper [38]. For these conditions periodic travelling waves have been observed. Two approaches are possible for simulating falling liquid films: imposing oscillating inlet flow conditions on a long computational domain resolving the spatial evolution of the waves or using spatially periodic boundary conditions to simulate one wavelength of the flow. Here, the second approach is chosen for ease of simulation thus imposing the wavelength rather than the frequency. Therefore, the computational domain length is equal to the wavelength and periodic boundaries are used in axial direction.

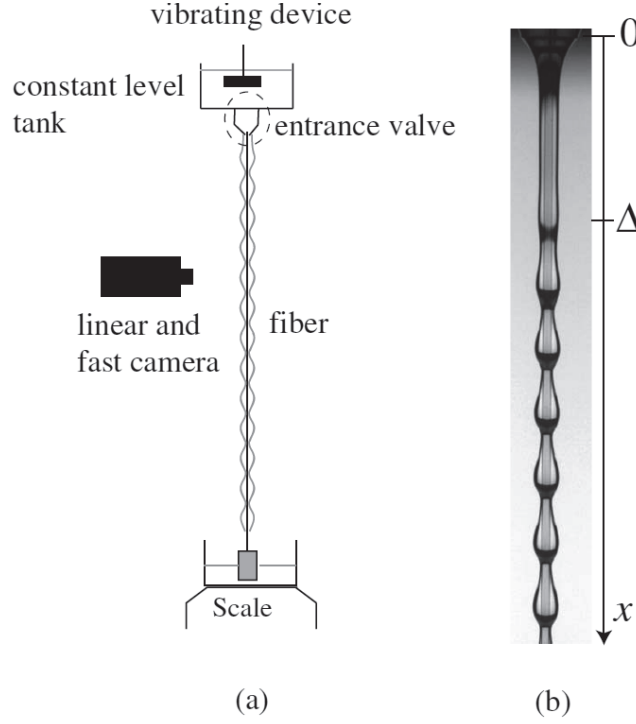


Figure 5.20: (a) Experimental setup used in [38], (b) photograph of the wavy film [38].

Air properties are used for the outer fluid and liquid properties are those of silicon oil v100. All lengths are non-dimensionalized by the fiber radius $R = 0.2mm$ and the velocity by $(10gR^2/\nu)$, where $g = 9.81m/s^2$. The interface is initialized with a sinusoidal perturbation of finite amplitude. Choosing different values of the initial amplitude was found to have little influence on the saturated shape. The amplitude first grows then saturates, due to non-linear effects, leading to a travelling wave of constant shape and celerity. The saturated wave shape is displayed in Fig. 5.21 where it is compared to the travelling wave shape computed from the weighted residual model of [38]. The simulation results in Fig. 5.21 are mesh converged. Besides a small vertical shift Fig. 5.21 displays close agreement between the weighted residual model of [38] and the direct numerical simulation thus suggesting that a two equation model can accurately capture the dynamics of waves on falling liquid films. The periodic boundary conditions used in the present simulation result in a constant volume of liquid inside the domain, which is prescribed at the beginning of the simulation. In the experiment on the other hand and in the weighted residual model solutions, the average liquid flow rate $\int_T q(x, t) dt$ (where T is the period) has been imposed. This could explain the slightly different liquid volume between weighted residual model and the numerical simulation in Fig. 5.21.

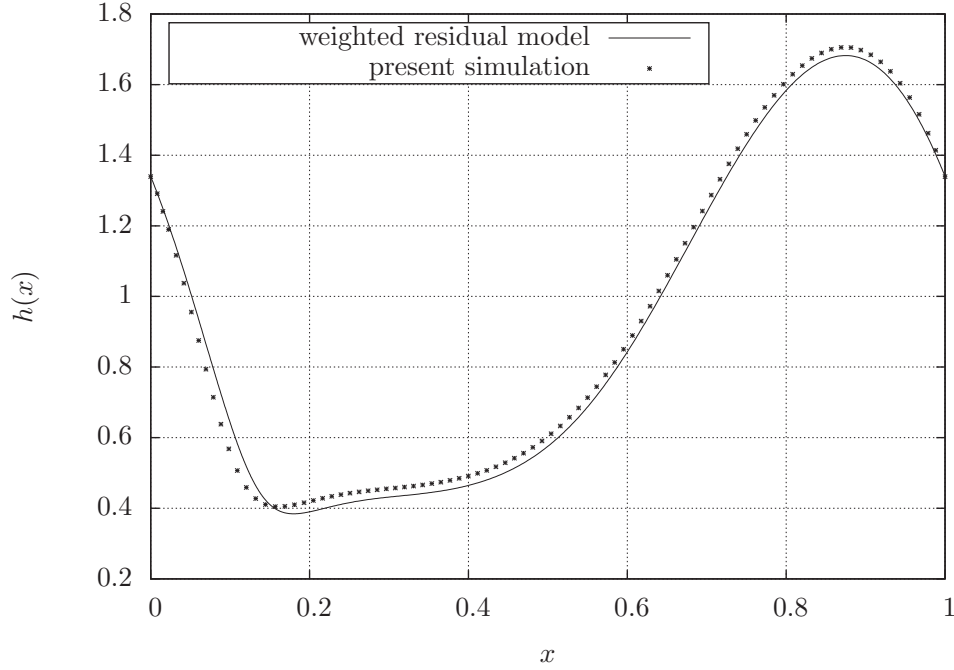


Figure 5.21: Travelling wave shape: film thickness h versus axial distance x . The present simulation is compared to the weighted residual model of [38].

5.10 Microchannel Simulations

In this section, simulations for an air bubble flowing in a microchannel are presented. The results are then compared to those found in [79], where bubble shapes and mass flow rates were obtained experimentally. Figure 5.22 shows the experimental facility used in [79]. A pressure pump was used to generate a flow of liquid, where air bubbles were injected through a T-junction located far enough upstream to ensure fully developed flow conditions in the measuring section. A micro-flow meter and a digital balance (placed at the channel outlet) were used to estimate the bulk flow rate of the liquid phase. Local flow measurements were performed using a micro-particle-shadow-velocimetry (μ PSV) technique, which consists in seeding the flow with particles of diameter $1.5\mu\text{m}$ and recording the particle shadows while they are illumination from the opposite side. The contrast due to the significant difference of refractive indices of the phases was then used to detect the phase interface on the shadowgraphy images.

Four cases from [79] are reproduced here spanning the range of parameters investigated in that paper. A schematic of the bubble's initial shape and the boundary conditions is displayed in Fig. 5.23. In a good approximation to the experimental conditions, a fully developed parabolic velocity profile is imposed at the inlet and at the outlet. The non-dimensional parameters Re and We are defined based on the channel diameter and the mean inflow velocity from the experiments. The pressure is set in the upper right corner in order to get a unique solution.

The geometry is chosen according to [79], by fixing R and calculating b such that the volume of

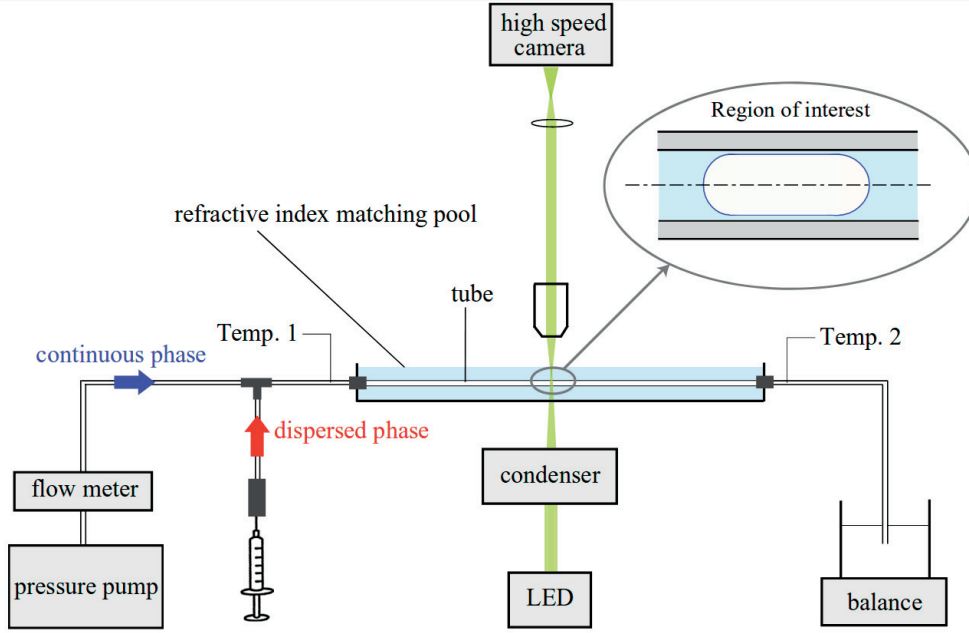


Figure 5.22: Experimental setup used in [79].

the bubble matches the experimental volume. In [79] two regimes were observed depending on the equivalent bubble diameter d_{eq} , which is the diameter of a spherical bubble with the same volume. For low d_{eq} the bubble remains close to spherical and the channel wall has little influence on the bubble shape, while for high d_{eq} elongated bubbles are observed.

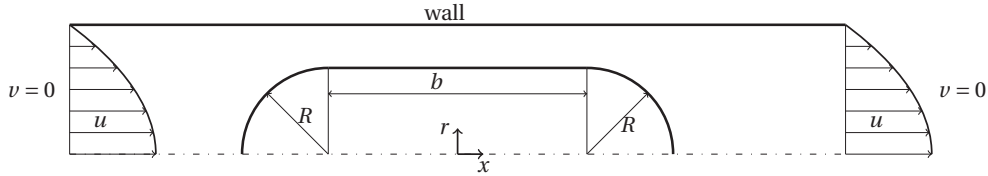


Figure 5.23: Schematic of the initial bubble geometry and boundary conditions.

We first consider the experiments with a glycerol solution as continuous phase. For an air bubble in glycerol the density and viscosity ratios are $9.6 \cdot 10^{-4}$ and $3.3 \cdot 10^{-5}$ respectively. The experimental pipe has a tube diameter of $494 \mu m$. Figure 5.24a displays the terminal bubble shape for an equivalent diameter of $d_{eq} = 0.677$, $Re = 0.004$ and $We = 1.3 \cdot 10^{-4}$. Only the upper half of the bubble is displayed and the axes are scaled according to the dimensions in the experiment. The channel wall is also plotted as a thick continuous black line. Due to the low Weber number the surface tension allows only small deviations from a spherical shape. The terminal bubble shape for a larger equivalent diameter of $d_{eq} = 1.047$, $Re = 0.005$ and $We = 1.8 \cdot 10^{-4}$ are shown in Fig. 5.24b. Here the bubble takes an elongated shape due to confinement.

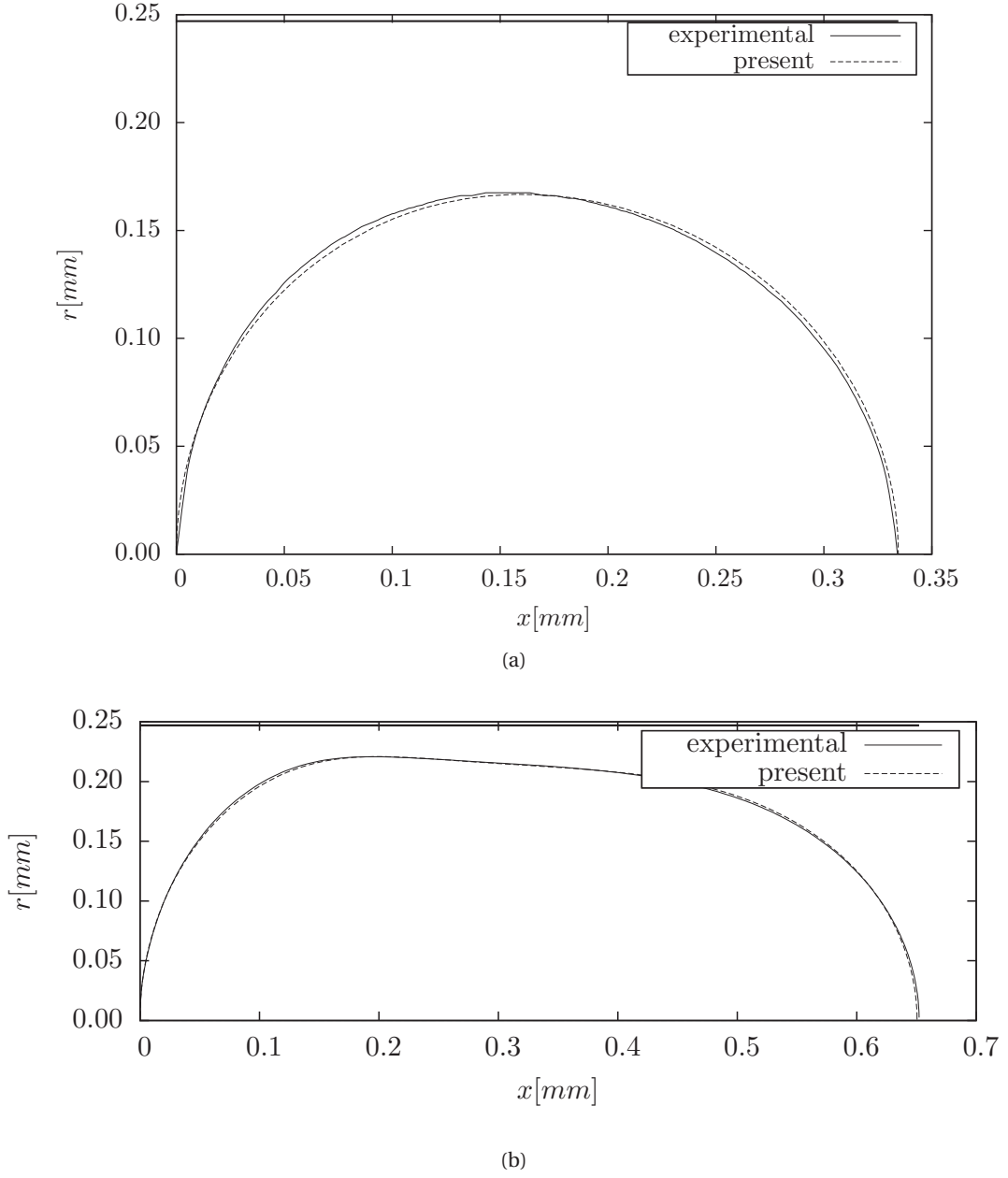


Figure 5.24: Terminal bubble shapes for an air bubble flowing in a microchannel filled with a glycerol solution. The equivalent diameter is $d_{eq} = 0.677$ (a) and $d_{eq} = 1.047$ (b).

We now consider another case with $d_{eq} > 1$ but now simulate the problem with a reference frame moving with the final bubble velocity in order to shorten the computational domain length. The distance between the bubble and the inlet/outlet boundaries is about 3 channel diameters and the bubble remains more or less in the middle of the domain. Choosing larger distances to the open boundaries was verified to have no influence on the results. The non-dimensional numbers are given by $Re = 0.01286$ and $We = 0.001278$. In Fig. 5.25a, the terminal

bubble shapes from the present simulations are compared to the experimental data and to the simulation results presented in [79] using the commercial software ANSYS Fluent. The simulations in [79] were carried out using an improved version of ANSYS Fluent based on the VOF method for interface advection and a height function curvature calculation. The steady state streamlines are also displayed in Fig. 5.25b. Good agreement is observed in Fig. 5.25a between the present simulations and those of [79] as well as between simulations and experiment. The simulations predict a steady state bubble velocity of 0.0179 m s^{-1} against a value of 0.01888 m s^{-1} in the experiment. This gives a relative error of about 5%, which is similar to the uncertainty range of the experimental measurement technique.

We now consider experiments with water for the liquid phase. The channel diameter is $514 \mu\text{m}$ and the non-dimensional numbers are given by $Re = 140.93$ and $We = 0.41225$. In this case the final bubble shape has a very thin liquid film with a thickness which is roughly 1% of the channel diameter. This makes the simulation more challenging as the thin film between the bubble and the wall has to be resolved by the mesh. In the present simulations a minimum of 5 mesh points were used to resolve the velocity profile in the gap between the bubble and the wall. The results for the air bubble in water case are displayed in Fig. 5.26a. In this case, the simulated terminal velocity was 0.254 m s^{-1} against 0.261 m s^{-1} in the experiment, yielding a relative error which is smaller than 3%. In Fig. 5.26a, the present results display better agreement with the experimental bubble shape than those of the competing simulations performed with the VOF method of ANSYS Fluent. Figure 5.26b displays profiles of the computed streamwise velocity taken at two different axial locations, where 'bub' is the velocity midway between the nose and tail of the bubble and 'chan' is the velocity in the channel sufficiently far from the bubble. Away from the bubble, the fully developed channel profile is recovered while at the bubble the velocity displays a parabolic profile in the gas region and a constant zero velocity is found in the thin liquid film. This is in accordance with Bretherton's small capillary number theoretical solution [20], which assumes a stagnant liquid film. For the results in Fig. 5.26, the capillary number is $Ca \approx 0.003$ and the leading order of the film thickness δ is given by [20]:

$$\frac{\delta}{R} = 0.643(3Ca)^{\frac{2}{3}} \approx 0.03,$$

which is very close to the film thickness observed in Fig. 5.26a.

5.11 Sessile Drop with static Contact Angle

In this section, we simulate the axisymmetric shape of a water droplet resting on a solid surface. We impose a static contact angle $\theta = \theta_s$ in the manner described in section 4.3. As mentioned earlier, the wetting phenomenon is a challenging problem in fluid mechanics. Here we check if the code is able to reproduce the steady solution and we choose $\theta_s = 90^\circ$. At a fixed contact angle, the steady shape of the droplet is determined by the Eötvös number ($Eo = We/Fr^2$) expressing the relative importance of gravity and surface tension forces. For small values of Eo the dominating surface tension force will try to maintain a constant curvature, while for

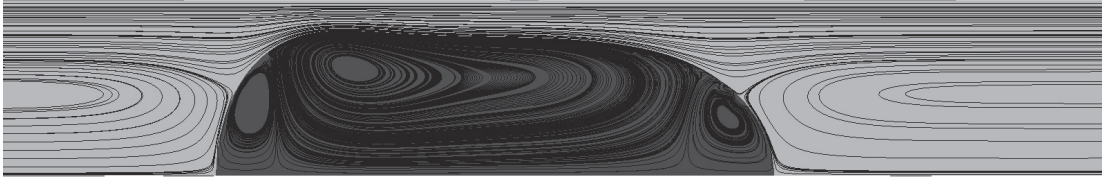
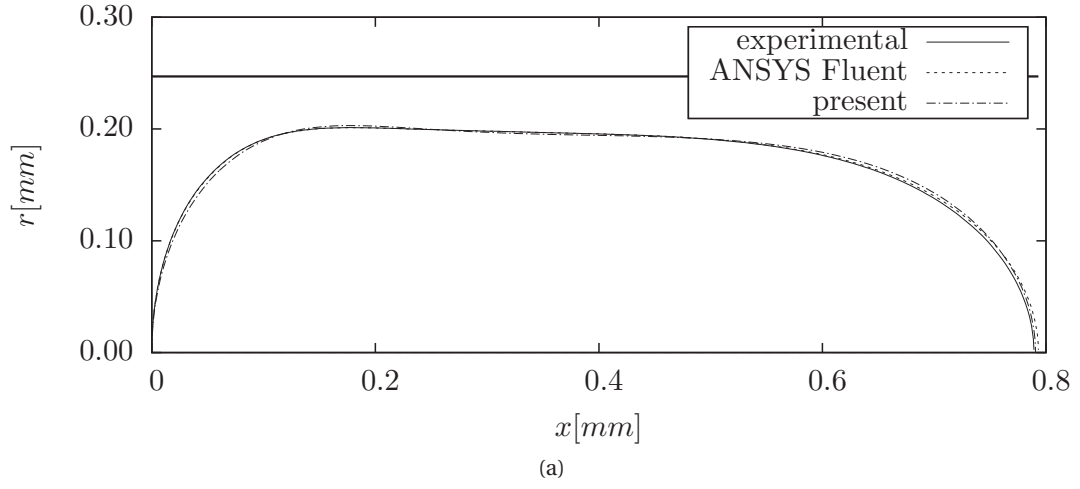


Figure 5.25: Final bubble shapes (a) and streamlines (b) for an elongated air bubble flowing in a microchannel filled with a glycerol solution.

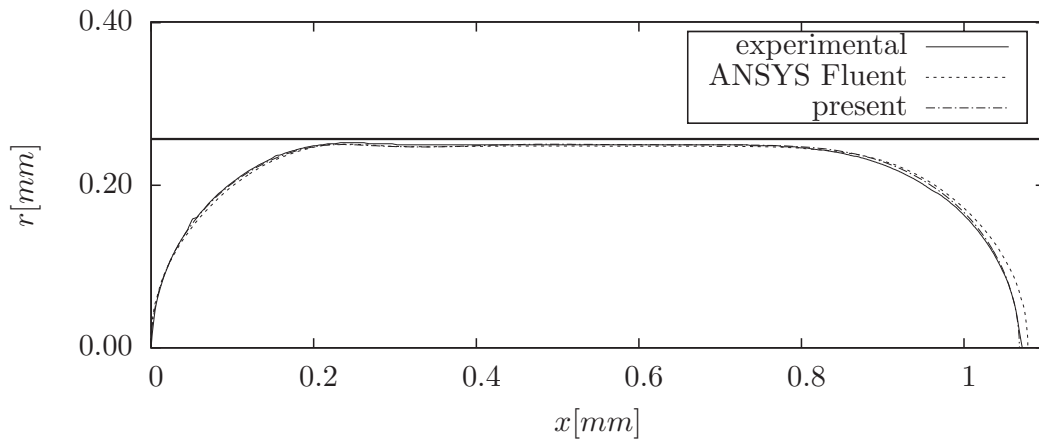
large values of Eo gravity will flatten the drop like a pancake.

For our simulation we chose a square computational domain of length $3R$ where we initialize a spherical drop with unit radius R . At the beginning the contact angle is imposed from Eq. (4.17) and we wait until the droplet shape does not change with time. The gravitational force acts in negative axial direction. The material properties are those of a water droplet surrounded by its vapor:

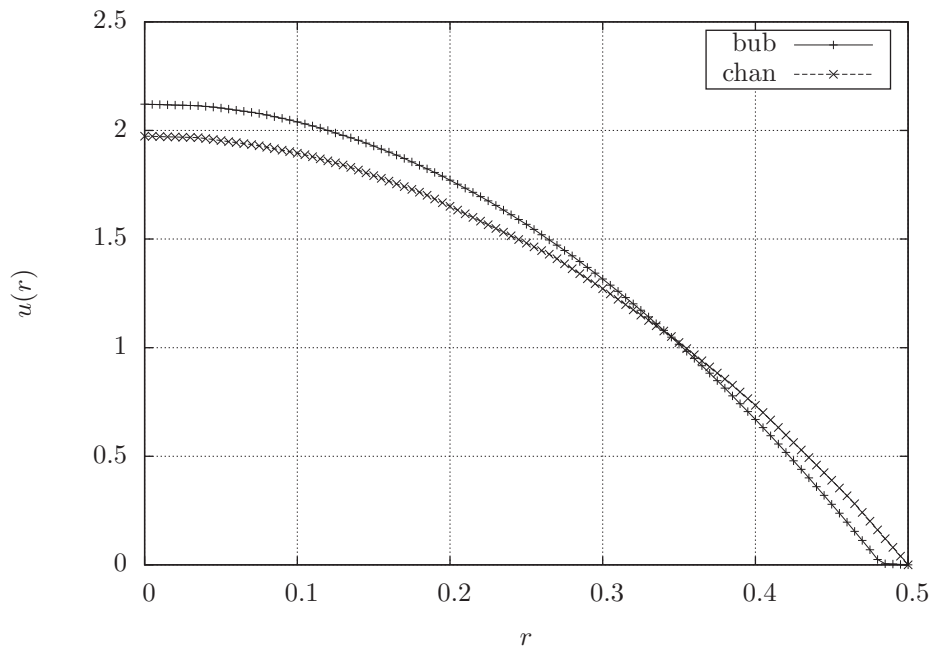
$$\frac{\rho_l}{\rho_v} = \frac{958.3}{0.597}, \quad \frac{\mu_l}{\mu_v} = \frac{28}{1.26}, \quad Re = 1, \quad We = 30, \quad Fr = \sqrt{\frac{We}{Eo}}$$

The final droplet shape for $Eo = 1$ and $Eo = 50$ are shown in Fig. 5.27. The figures confirm the effect of the Eötvös number: for $Eo = 1$ the strong surface tension leads to a droplet shape very close to a spherical cap, while for $Eo = 50$ the drop is flattened by gravity.

The shape of a static droplet resting forming a given contact angle θ_s with the substrate can be computed by integrating the Young Laplace equation. The Young Laplace equation yields,



(a)

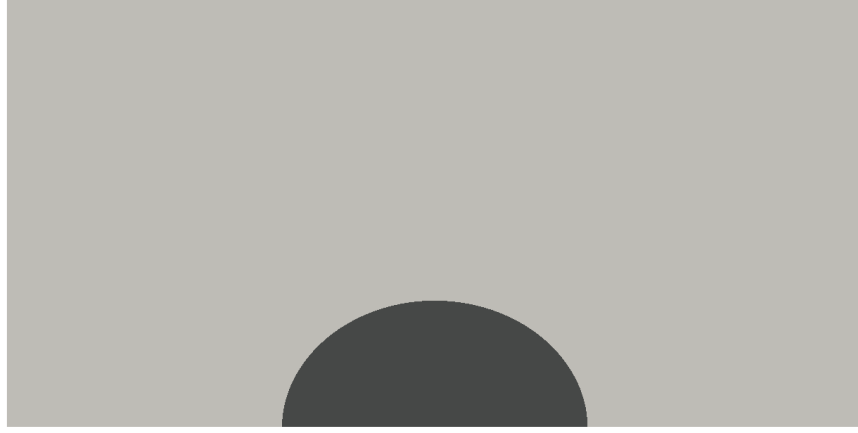


(b)

Figure 5.26: Final bubble shapes (a) and non-dimensional velocity profiles (b) for an elongated air bubble flowing in a microchannel filled with water.

which equals the capillary pressure difference to the hydrostatic pressure, reads as:

$$\frac{\rho_l(x_0 - x)}{Fr^2} = \frac{\kappa}{We},$$



(a)



(b)

Figure 5.27: Steady droplet shapes for $\theta_s = 90^\circ$ and (a) $Eo = 1$, (b) $Eo = 50$. The axis of symmetry is vertical.

where κ is two times the mean curvature and the pressure in the ambient vapor is assumed to vanish. Inserting the expression for the axisymmetric curvature gives:

$$\frac{2}{R} - zEo + \frac{z''}{(1 + z'^2)^{3/2}} + \frac{z'}{r} \frac{1}{\sqrt{1 + z'^2}} = 0, \quad (5.14)$$

where $z = x - h$, $z' = dz/dr$, h is the final height and R is the curvature radius at the apex of the drop. Equation (5.14) is formulated in terms of the angle $\phi = \tan^{-1}(z')$:

$$\frac{dr}{d\phi} = \frac{r \cos \phi}{r(Eo z - 2/R) - \sin \phi}, \quad \frac{dz}{d\phi} = \frac{r \sin \phi}{r(Eo z - 2/R) - \sin \phi}.$$

It can then be integrated from the apex where $\phi = 0$ to the surface where $\phi = -\theta_s$. In order to compare with the numerical solution we vary R in Eq. (5.14) until the volume of the droplet matches the volume in the numerical simulation. The droplet shapes obtained with this theoretical approach are compared to the numerical simulations in Fig. 5.28 for three different values (1, 10, 50) of Eo and good agreement is observed.

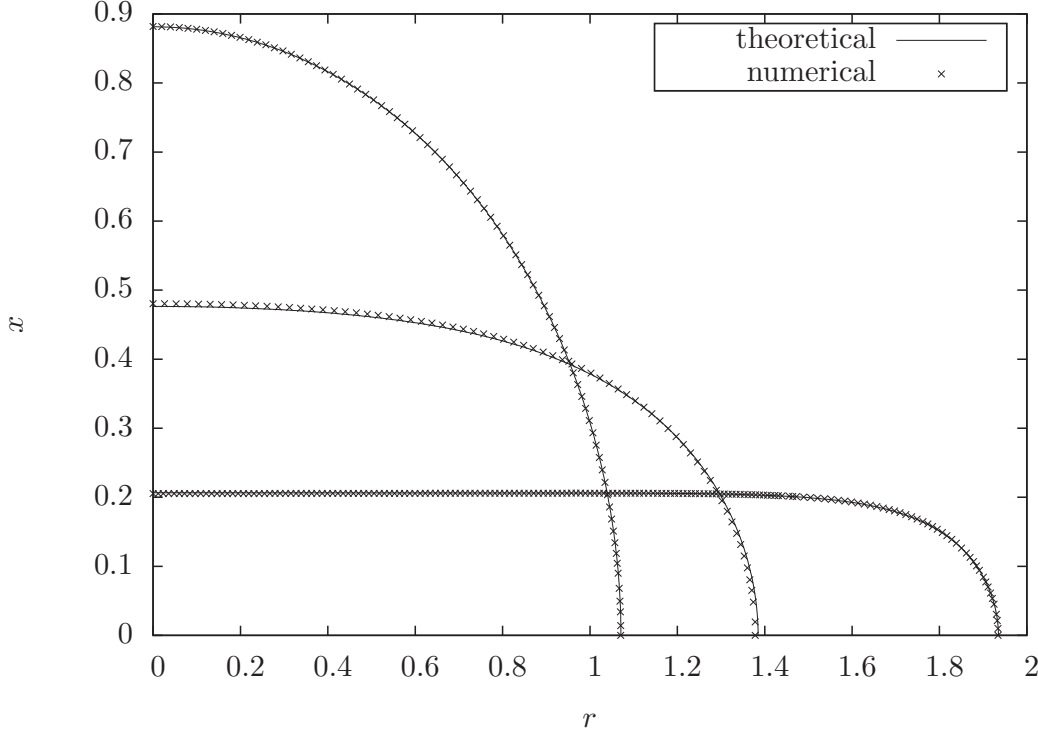


Figure 5.28: Theoretical and numerical droplet shapes for $Eo = 1, 10, 50$ (top to bottom).

5.12 1D Phase Change Problem

This section verifies the phase change implementation by simulating a classical 1D Stefan problem. The problem geometry is depicted in Fig. 5.29. A similar setup has been used by several authors [43, 60, 114, 132]. The vapor is in contact with a hot wall and the liquid is at saturation temperature. Therefore, the boundary conditions are $T(x = 0, t) = T_{wall}$, $T(x \geq X, t) = T_{sat}$, where $X(t)$ is the advancing interface position. The dimensionless temperature $T(x, t)$ is defined as the deviation from saturation conditions such that $T_{sat} = 0$ and a constant superheated temperature $T_{wall} = 1$ is used at the wall. Evaporation occurs due to the heat transferred from the wall through the vapor phase. The velocity vanishes everywhere in the vapor phase since the latter is confined by the wall. This gives the following relation between

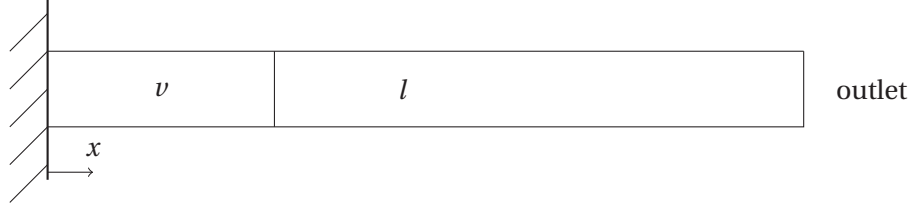


Figure 5.29: Schematic of the 1D Stefan problem.

the interface velocity \dot{X} and the heat flux in the vapor:

$$\rho_v \dot{X} = \dot{m}'' = \frac{\lambda_v}{\Delta h_{lv} Re Pr} \frac{\partial T}{\partial x} \Big|_X. \quad (5.15)$$

An analytical solution can be derived by introducing a similarity variable $s = \frac{x}{2\sqrt{\alpha_v t}}$, where $\alpha_v = \frac{\lambda}{\rho c_p Re Pr}$ is the dimensionless thermal diffusivity of the vapor. Since there is no flow inside the vapor, the energy equation reads

$$\frac{DT}{Dt} = \frac{\partial T}{\partial t} = \alpha_v \frac{\partial^2 T}{\partial x^2}. \quad (5.16)$$

The interface position evolves as the square root of time:

$$X(t) = 2\beta\sqrt{\alpha_v t}, \quad (5.17)$$

where β is a root of the following transcendental equation:

$$\beta e^{\beta^2} \operatorname{erf}(\beta) = \frac{c_p(T_{wall} - T_{sat})}{\Delta h_{lv}\sqrt{\pi}}. \quad (5.18)$$

The liquid is at saturation temperature and the temperature in the vapor ($x < X$) reads:

$$T(x, t) = T_{wall} - (T_{wall} - T_{sat}) \frac{\operatorname{erf}(\frac{\beta x}{X(t)})}{\operatorname{erf}(\beta)}. \quad (5.19)$$

A triangular mesh initially consisting of two horizontal layers of triangles was used for the simulations, with symmetry boundary conditions on the horizontal boundaries. The initial edge size of the mesh was 0.005 and the initial interface position was $X(0) = 0.1$. The mesh motion parameters in Eq. (4.30) were chosen as $\beta_1 = 0, \beta_2 = 0.8$. Fluid properties corresponding to refrigerant R134a were used for the simulation: $\rho_l = 1187 \text{ kg m}^{-3}$, $\lambda_l = 0.079 \text{ W m}^{-1} \text{ K}^{-1}$, $c_{pl} = 4.26 \cdot 10^{-4} \text{ J kg}^{-1} \text{ K}^{-1}$ and $\rho_v = 37.54 \text{ kg m}^{-3}$, $\lambda_v = 0.0173 \text{ W m}^{-1} \text{ K}^{-1}$, $c_{pv} = 4.26 \cdot 10^{-4} \text{ J kg}^{-1} \text{ K}^{-1}$. The non-dimensional numbers were all set to unity $Re = We = Pr = \Delta h_{lv} = 1$ and (5.18) gives $\beta = 0.620063$. The computed interface position is compared to the theoretical solution given by Eq. (5.17) in Fig 5.30a and the temperature profiles are compared to Eq. (5.19) in Fig 5.30b. Good agreement is found between numerical and theoretical results.

5.13 Evaporating Bubble in Superheated Liquid

In this section we investigate a spherical bubble expanding in a uniformly superheated, unbounded pool of liquid. The liquid is superheated with $T(r = \infty, t) = T_\infty > T_{sat}$ and the vapor bubble is at saturation temperature $T(r \leq R, t) = T_{sat}$, where r is the polar radius and $R(t)$ is the time dependent bubble radius. This idealized problem is relevant to the study of nucleate boiling, where two stages of bubble growth can be distinguished [24]: the inertia-controlled growth and the heat-transfer-controlled growth. The inertia-controlled growth exists during the early stages of a nucleated vapor bubble. Here we consider the heat-transfer-controlled growth, which is found in the later stages when growth is limited by the transport of heat to the interface. Scriven derived a theoretical solution [120], which is useful for verification purposes. Scriven's solution is briefly introduced here. Similarly to the previous case, the vapor phase is confined and the velocity inside the bubble has to vanish. To fulfil mass conservation, the liquid velocity has to decay as the inverse of the square of the radius:

$$v(r) = V_R \frac{R^2}{r^2}, \quad (5.20)$$

where r is the spherical radius and $V_R = v(r = R)$. The latter can be eliminated by using the interface mass balance condition:

$$\dot{m}'' = \rho_v \dot{R} = \rho_l (\dot{R} - V_R) \Rightarrow V_R = \dot{R} \left(1 - \frac{\rho_v}{\rho_l} \right). \quad (5.21)$$

Using all the assumptions listed in [120], the energy equation in the liquid surrounding the bubble reads as:

$$\frac{\partial T}{\partial t} + \epsilon \dot{R} \frac{R^2}{r^2} \frac{\partial T}{\partial r} = \alpha_l \left(\frac{2}{r} \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial r^2} \right), \quad (5.22)$$

where $\epsilon = \left(1 - \frac{\rho_v}{\rho_l} \right)$. The bubble radius grows proportional to the square root of time:

$$R(t) = 2\beta \sqrt{\alpha_l t}, \quad (5.23)$$

and with the substitution $s = \frac{r}{2\sqrt{\alpha_l t}}$, Eq. (5.22) becomes

$$\frac{d^2 T}{ds^2} + 2 \left(s + \frac{1}{s} - \frac{\epsilon \beta^3}{s^2} \right) \frac{dT}{ds} = 0. \quad (5.24)$$

Integrating Eq. (5.24) twice, taking into account the boundary condition $T(s = \infty) = T_\infty$, gives

$$T(s) = T_\infty - A \int_s^\infty \frac{\exp(-x^2 - 2\epsilon \beta^3 x^{-1})}{x^2} dx \quad (5.25)$$

5.13. Evaporating Bubble in Superheated Liquid

Using the remaining boundary conditions, the constants A and β are calculated as follows:

$$\begin{aligned} \dot{m}'' &= \rho_v \dot{R} = \rho_v \beta \sqrt{\frac{\alpha_l}{t}} = \frac{\lambda_l}{\Delta h_{lv} Re Pr} \frac{\partial T}{\partial r} \Big|_R, \\ \Rightarrow A &= 2 \Delta h_{lv} c_p \frac{\rho_v}{\rho_l} \beta^3 \exp((1 + 2\epsilon)\beta^2) \end{aligned} \quad (5.26)$$

and

$$T(r = R, t) = T_{sat} = T_\infty - A \int_{\beta}^{\infty} \frac{\exp(-s^2 - 2\epsilon\beta^3 s^{-1})}{s^2} ds. \quad (5.27)$$

Equations (5.25) and (5.27) require the evaluation of an improper integral. These improper integrals were computed by splitting the limits

$$\int_a^{\infty} f(x) dx = \int_a^1 f(x) dx + \int_1^{\infty} f(x) dx \quad (5.28)$$

and transforming the second integral with the substitution $u = 1/x$:

$$\int_1^{\infty} \frac{\exp(-x^2 - 2\epsilon\beta^3 x^{-1})}{x^2} dx = \int_0^1 \exp(-u^{-2} - 2\epsilon\beta^3 u) du. \quad (5.29)$$

The resulting integrals are of standard type and can be computed by Gaussian quadrature. Scriven's solution gives the time dependent bubble radius (5.23), the velocity distribution (5.20) and the temperature distribution (5.25).

An axisymmetric domain was used in the simulations taking advantage of the spherical symmetry of the problem. The non-dimensional parameters and fluid properties assumed the following values:

$$\begin{aligned} Re &= 1, \quad We = 30, \quad Pr = 1, \quad R(t=0) = 0.5, \quad T_\infty = 1, \quad T_{sat} = 0, \quad \Delta h_{lv} = 60000, \\ \rho_l &= 958.3 \text{ kg m}^{-3}, \quad \mu_l = 2.8 \cdot 10^{-4} \text{ kg m}^{-1} \text{ s}^{-1}, \quad \lambda_l = 0.679 \text{ W m}^{-1} \text{ K}^{-1}, \quad c_{pl} = 2425 \text{ J kg}^{-1} \text{ K}^{-1}, \\ \rho_v &= 0.597 \text{ kg m}^{-3}, \quad \mu_v = 1.26 \cdot 10^{-5} \text{ kg m}^{-1} \text{ s}^{-1}, \quad \lambda_v = 0.025 \text{ W m}^{-1} \text{ K}^{-1}, \quad c_{pv} = 2425 \text{ J kg}^{-1} \text{ K}^{-1}. \end{aligned}$$

The parameter β is calculated to be 0.128257. Figure 5.31 displays the computational domain and the mesh. The horizontal boundary of the domain is a symmetry line and an outflow boundary condition, with a fixed value of the pressure and temperature: $p = 0, T = T_\infty = 1$, is imposed on the circular part of the boundary. The temperature inside the bubble is constant and equal to the saturation temperature, which is given by $T = 0$. In the liquid outside the bubble, the temperature is initialized with the values given by the theoretical solution (5.25). The mesh motion parameters were set to $\beta_1 = 0, \beta_2 = 0.8$ and simulations were performed

with different mesh and time step sizes to obtain converged results.

The computed bubble radius is compared to the theoretical solution in Fig 5.32. The bubble growth, proportional to the square root of time, predicted by Eq. (5.23) is closely matched. Figure 5.33 compares the radial velocity and the temperature to the exact solution. A steep temperature boundary layer is observed close to the bubble as evidenced in Fig. 5.33b. The mesh displayed in Fig. 5.31 is strongly refined close to the bubble allowing it to resolve the boundary layer. Overall good agreement is found between the numerical and the theoretical solution.

5.14 Rayleigh-Taylor Instability with Phase Change

This last test-case demonstrates the effects of phase change on the Rayleigh-Taylor instability. It has been found [65] that phase change can reduce the growth rate of infinitesimal perturbations around a base flow with a constant temperature gradient. A planar geometry is considered where the liquid occupies $0 < y < 1$ and the vapor $-1 < y < 0$. The quiescent base flow has a linear temperature profile:

$$\begin{aligned} T &= G_l y \quad \text{for } 0 < y < 1, \\ T &= G_v y \quad \text{for } -1 < y < 0, \end{aligned}$$

where the gradients are such that there is no phase change in the base flow: $\lambda_l G_l = \lambda_v G_v$. The flow parameters correspond to those used in [44]:

$$\begin{aligned} Re &= 2500, \quad We = 8.333, \quad Fr = 0.289, \\ Pr &= 0.00947, \quad T_{sat} = 0, \quad \Delta h_{lv} = 0.065, \\ \rho_l &= 2.5 \, \text{kgm}^{-3}, \quad \mu_l = 0.001 \, \text{kgm}^{-1}\text{s}^{-1}, \quad \lambda_l = 0.1056 \, \text{Wm}^{-1}\text{K}^{-1}, \\ \rho_v &= 0.25 \, \text{kgm}^{-3}, \quad \mu_v = 0.0001 \, \text{kgm}^{-1}\text{s}^{-1}, \quad \lambda_v = 0.0211 \, \text{Wm}^{-1}\text{K}^{-1}. \end{aligned}$$

A sinusoidal perturbation with a unit wave length is applied to the interface: $y = A_0 \cos(2\pi x)$ with $A_0 = 0.0005$. The width of the domain is a single wavelength. The problem is first simulated without phase change, where the classical dispersion relation:

$$\omega^2 = (\sigma k^3 - \Delta \rho k g) / (\rho_l + \rho_v) \tag{5.30}$$

should hold. In a second step, phase change is included and a reduced growth rate is observed in accordance with theory. The instantaneous amplitude of the perturbation $A(t)$ is plotted in Fig. 5.34, with a logarithmically scaled ordinate such that exponential growth appears as a constant slope. The exponential growth from linear stability analysis is represented by solid lines in Fig. 5.34. After an initial adjustment period, the numerical growth rates are close to those from linear stability analysis. The results in Fig. 5.34 are very similar to the computations of [44].

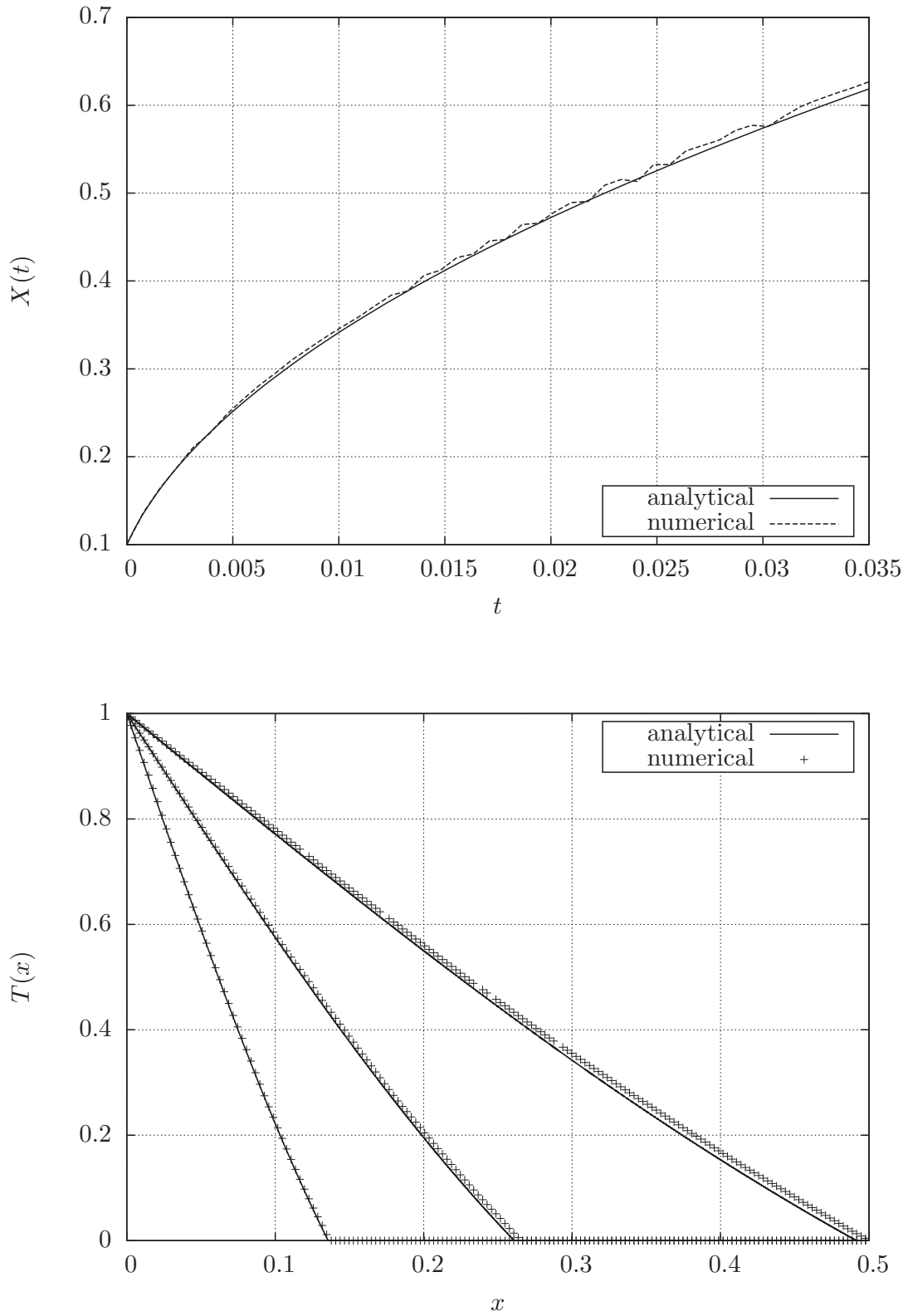


Figure 5.30: (a) Interface position as a function of time, (b) temperature distribution at several instants $t = 0.00079, 0.0055, 0.022$ (from left to right).

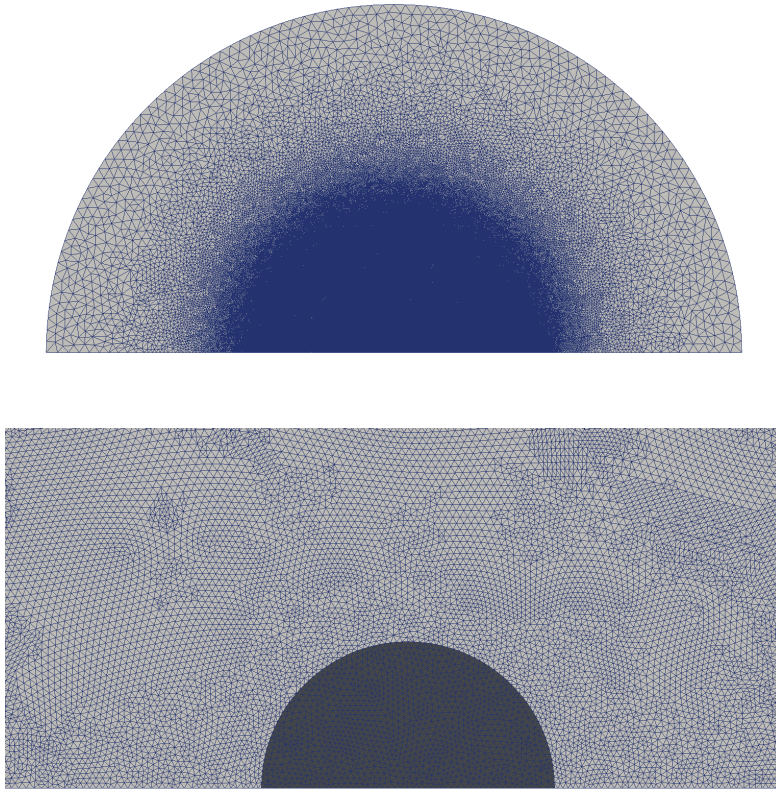


Figure 5.31: Meshed computational domain (a), zoom on the bubble (b).

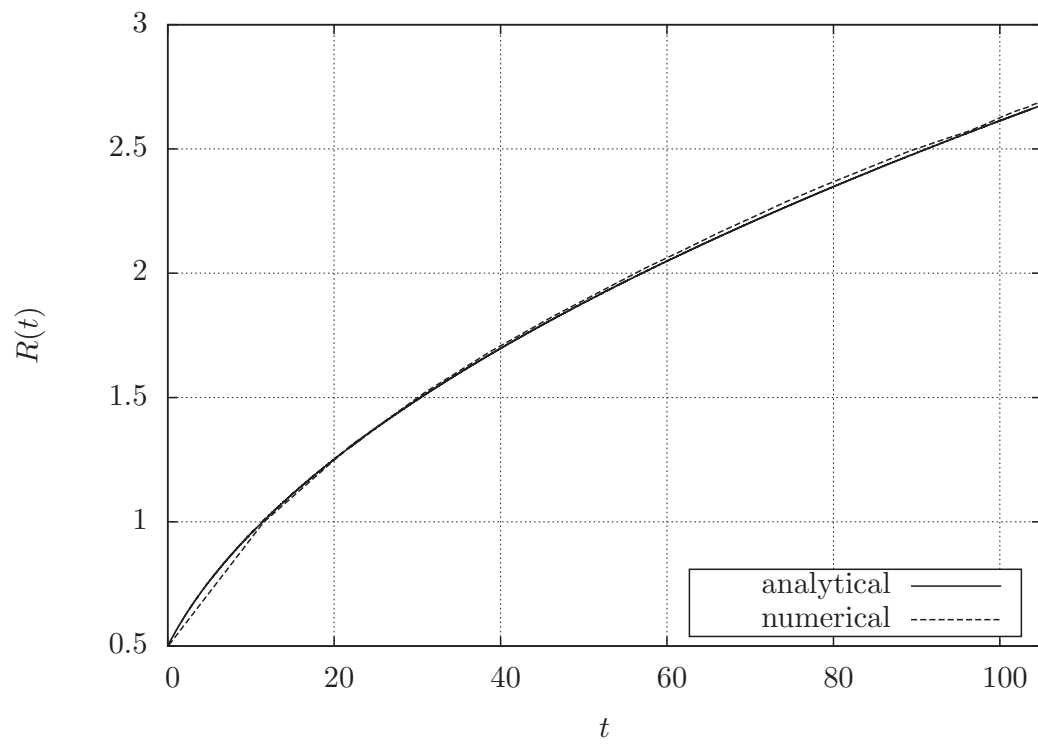


Figure 5.32: Bubble radius as a function of time.

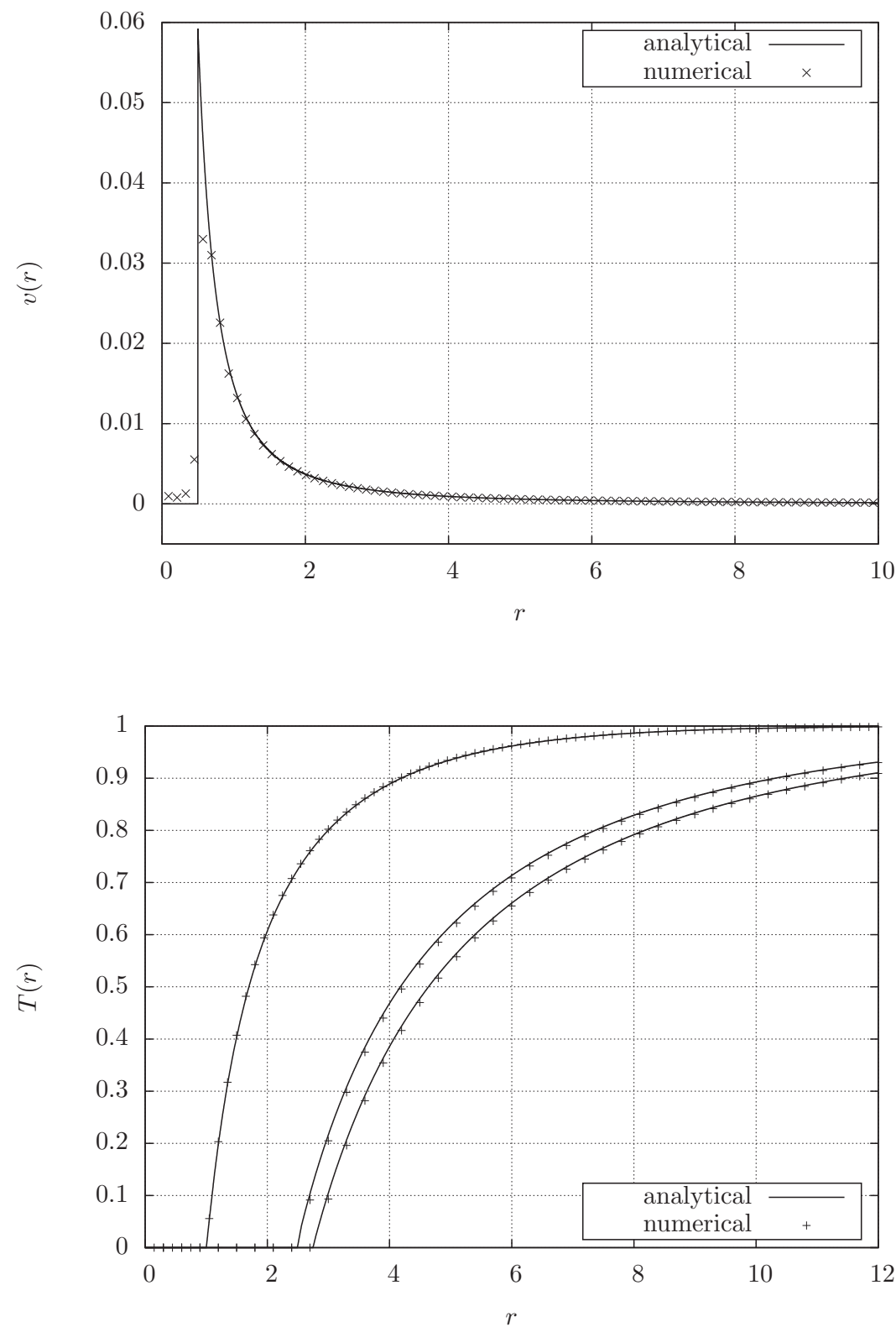


Figure 5.33: Radial distribution of (a) initial radial velocity and (b) temperature at several instants $t = 11.5, 89, 110.5$ (from left to right).

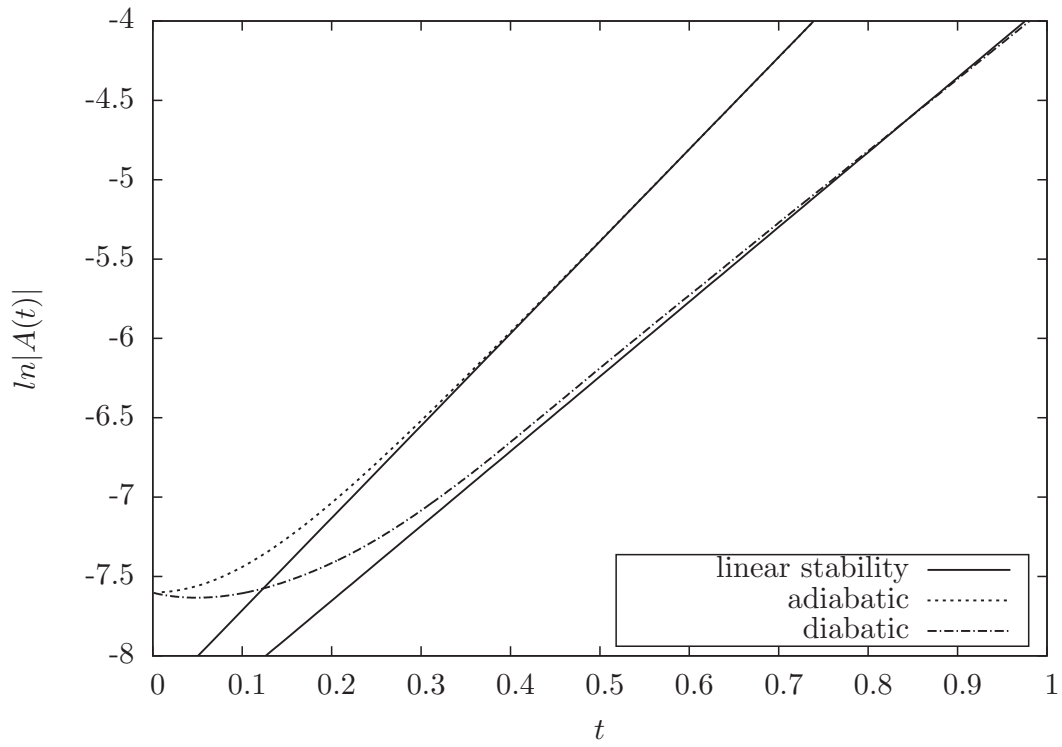


Figure 5.34: Evolution of the perturbation amplitude with (diabatic) and without (adiabatic) phase change.

6 Conclusions and Outlook

Our aim was to gain more understanding of the issues involved in numerical simulation of two-phase flows. The focus was on surface tension dominated problems and phase change. For this task a FEM discretization of the Navier-Stokes equations was used, with unstructured triangular meshes. The governing equations were solved on interface resolving meshes using the Arbitrary Lagrangian-Eulerian (ALE) description. Phase change was implemented via a source term in the continuity equation, with the rate of mass transfer computed from the interfacial jump in conductive heat flux.

In FEM computations, the regularity of the element shape affects the computational precision and the conditioning of the system matrix. Therefore, when working with moving and deforming meshes, it is important to monitor element size and shape at all times. If the element deformations become too large, remeshing has to be applied. In this thesis, the remeshing strategy is to insert and remove points based on the local element size recreate the mesh connectivity and smooth the element shape with the moving mesh velocities. Adaptive mesh refinement is thus possible with the present method by choosing the velocity used to update the location of the mesh nodes and by changing the number of mesh nodes. Mesh points are deleted, added or displaced and the solution is interpolated on the new mesh. The criterion to insert/remove mesh points is based on the ratio of actual edge length to a target edge length distribution, which is found by solving a Helmholtz equation. This adaptive approach allows to handle complicated mesh motion and accurately resolve fine scales of the flow. However, remeshing should be used with care since it is expensive and causes additional errors. Choosing appropriate mesh velocities helps to keep the remeshing frequency low.

The versatility of the approach has been demonstrated by several testcases from the two-phase flow literature. Cases with large density and viscosity ratios could be simulated accurately. Using a discrete version of the Frenet-Serret formula for the curvature calculation proved to be a good choice, as high accuracy could be achieved at very low computational cost. The accurate curvature estimation combined with a consistent discretization of the pressure gradient and surface tension term resulted in very low parasitic currents and accurate capillary pressure jumps. In the particular case of a spherical drop in a constant velocity field the errors

were on the order of the machine accuracy. The method was shown to correctly describe the frequency of capillary waves on a spherical droplet and to handle large deformations as they occur in the non-linear stages of a Rayleigh-Taylor instability. The shape of travelling waves in falling film flow around a fibre and growth rates from linear stability analysis were well captured. Phase change validations were performed for flat, spherical and sinusoidally perturbed interfaces. Simulation results were successfully compared to experimental data for the flow of an air bubble in a microchannel. Moreover, all computations in this thesis were run in serial on a desktop computer. It can thus be concluded that the ALE moving mesh method is accurate and efficient for the simulation of two-phase flows with heat and mass transfer. However, the present method has some limitations related to the lack of flexibility in dealing with complex interface deformation. Finding the proper choice of mesh parameters can be a difficult task, which needs to be performed on a case by case basis.

6.1 Perspectives

In the future the code could be used to study diverse two-phase physical phenomena. The present approach is especially suitable for problems where the interface has simple topology but a high degree of fidelity is required for its representation. Some examples of possible applications are: rising bubbles, droplet impact, evaporating droplets on substrates, evaporating channel flow, break up of liquid threads and bubble coalescence. For a droplet evaporating on a surface, under diffusional control, two extreme modes of evaporation have been identified [97]: the constant contact angle (CA) mode and the constant contact radius (CR) mode. In the theory of [97] the drop is assumed to be a spherical cap, in the first mode the contact radius shrinks while the contact angle is constant and in the second mode the contact angle reduces while the contact radius is pinned. While the CA and CR modes can be simulated with the present code real drops tend to spread with a dynamic contact angle that depends on the contact line motion. In [96], a dynamic contact angle model, with a contact line velocity dependent contact angle, was shown to capture the dynamics of a droplet impacting on a surface. An interesting application, which does not involve contact lines, is the Leidenfrost effect. A droplet impinging on a hot plate will enter the Leidenfrost regime when the wall temperature exceeds a certain limit. In this case the droplet levitates on a thin layer of saturated vapor between the droplet and the plate. Leidenfrost boiling was simulated in [113] using the level-set method to represent the interface, while applying the jump conditions at the interface using a Ghost Fluid approach based on a divergence-free extrapolation of the velocity field. Numerical computations presented in [113] resolved the boiling dynamics in the vapor layer with a very dense (static) mesh. The present adaptive refinement method could be used to resolve the thin vapor layer more efficiently.

Break up of a fluid treads and merging of spherical bubbles could be simulated with the present approach. However, as the interface is explicitly marked by connected mesh points, topological changes cannot happen by default. Such problems could be simulated up to the singularity or breakup/merging could be implemented via mesh separation/combination criteria, like

those proposed in [106, 107] where mesh separation and combination methods were used to simulate jet pinching and the off-center collision of droplets including the formation of satellite droplets. However, as was noted in [107] the time when mesh separation/combination are applied is ad-hoc and executed when radius of the neck is smaller than a length defined as a fraction of the initial drop diameter. Moreover, these schemes were implemented by converting several cells of one fluid in another thus leading to a mass conservation error. When a jet breaks up due to surface tension it forms a very thin and almost perfectly axisymmetric thread, whose dynamics near breakup are found to exhibit universal scaling laws [40]. Here the adaptive mesh refinement capability could be particularly useful to allow the dynamics of breakup to be simulated across the spatial range in the inertia and viscous regime.

6.1.1 Improvements to the Numerical Method

The Lagrangian interpolation described in section 4.10 is expected to represent a large source of error in the present approach. This type of interpolation does not conserve mass and energy and is unsuited for discontinuous discretizations [46]. An alternative Galerkin projection interpolation would improve on these deficiencies. Galerkin projection minimizes the integral over the target mesh of the squared interpolation error in the L_2 -norm. For a Galerkin projection interpolation it is suitable to perform integration over a "supermesh" i.e. a mesh containing all the nodes of both the original and the new mesh.

The implemented contact angle model, that was described in section 4.3, does not allow to use a velocity dependent contact angle. Moreover, if a contact angle different than $\pi/2$ is used it is unclear what boundary conditions to use for the curvature on the interface boundary. In order to remove these issues it would be advisable to include the curvature in the weak form. This would require a departure from the currently used "one fluid" approach where the surface tension is included as a body force via Eq. (3.19). However, since an interface adapted mesh is used, it is possible to use a "two fluids" formulation and impose the dynamic interface condition (3.11) at the interface. The curvature boundary condition and the dynamic contact angle model could then be introduced naturally into the boundary terms resulting from integration by parts of the stress tensor. The Frenet-Serret formula for the curvature (4.12) can also be integrated by parts transferring the derivative to the test function and resulting in a boundary term similar to equation (2.25) of [6]. An arbitrary contact angle value could then be prescribed by specifying the direction of the unit tangent vectors at the endpoints of the interface.

Parallelizing the code is advisable if larger systems are going to be simulated. A shared data parallelization (based on OpenMP) could be a first step, which was already started during this thesis but only for the FEM matrix assembly. Most of the computational effort goes into solving the linear system wherefore a great speedup could be achieved by parallelizing it. A distributed data MPI parallelization is expected to be necessary for higher performance but remeshing will require repeated domain decompositions or complicated load balancing

Chapter 6. Conclusions and Outlook

procedures making this a non-trivial task.

A Coordinate Transformations of the Navier-Stokes Equations

Since the governing equations for axisymmetric flow play a central role in this thesis, the derivation of the cylindrical Navier-Stokes equations is carried out in this appendix. We start by giving general expressions for the transformation of the individual terms in the Navier-Stokes equations from Cartesian into arbitrary curvilinear coordinate systems. These formulas are then applied to the cylindrical case but they are equally useful to obtain other forms of the equations, for example the spherical case.

A.1 From Cartesian to Curvilinear Coordinates

Starting point are the incompressible Navier-Stokes equations in Cartesian coordinates (x_i) . In order to avoid the transformation of tensors, which is more involved, we start from the non-conservative velocity pressure formulation:

$$\begin{aligned} \rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) &= -\nabla p + \mu \Delta \mathbf{v} + \rho \mathbf{g}, \\ \nabla \cdot \mathbf{v} &= 0, \end{aligned} \tag{A.1}$$

where

$$\mathbf{v} = \sum_i \mathbf{e}_i v_i, \tag{A.2}$$

$$\nabla = \sum_i \mathbf{e}_i \frac{\partial}{\partial x_i}, \tag{A.3}$$

$$\Delta = \sum_i \frac{\partial^2}{\partial x_i^2}. \tag{A.4}$$

The vectors \mathbf{e}_i form an orthonormal basis in the Cartesian coordinate system.

Appendix A. Coordinate Transformations of the Navier-Stokes Equations

Next some general formulas are going to be given, which allow to express the Navier-Stokes equations (A.1) in another coordinate system with coordinates (\tilde{x}_i) .

In the new coordinate system, the unit basis vectors are defined as

$$\tilde{\mathbf{e}}_i = \frac{\partial \mathbf{r}}{\partial \tilde{x}_i}, \quad (\text{A.5})$$

and the velocity vector is expressed as

$$\mathbf{v} = \sum_i \tilde{\mathbf{e}}_i \tilde{v}_i. \quad (\text{A.6})$$

It follows from that last equation together with the chain rule:

$$\frac{\partial}{\partial x_i} = \sum_j \frac{\partial \tilde{x}_j}{\partial x_i} \frac{\partial}{\partial \tilde{x}_j}, \quad (\text{A.7})$$

that

$$v_i = \sum_j \tilde{v}_j \frac{\partial x_i}{\partial \tilde{x}_j}. \quad (\text{A.8})$$

Therefore, divergence of the velocity is

$$\nabla \cdot \mathbf{v} = \sum_i \frac{\partial v_i}{\partial x_i} = \sum_i \sum_j \sum_k \frac{\partial \tilde{x}_i}{\partial x_k} \frac{\partial}{\partial \tilde{x}_i} \left(\tilde{v}_j \frac{\partial x_k}{\partial \tilde{x}_j} \right). \quad (\text{A.9})$$

after transformation.

The convective term in the momentum equation, transforms as

$$\mathbf{v} \cdot \nabla \mathbf{v} = \sum_i \sum_j \sum_k \tilde{v}_j \frac{\partial x_k}{\partial \tilde{x}_j} \frac{\partial \tilde{x}_i}{\partial x_k} \frac{\partial}{\partial \tilde{x}_i} \left(\sum_l \tilde{\mathbf{e}}_l \tilde{v}_l \right) = \sum_i \tilde{v}_i \frac{\partial}{\partial \tilde{x}_i} \left(\sum_l \tilde{\mathbf{e}}_l \tilde{v}_l \right). \quad (\text{A.10})$$

From (A.3) it follows that the gradient operator can be expressed as

$$\nabla = \sum_i \tilde{\mathbf{e}}_i \sum_j S_{ij} \frac{\partial}{\partial \tilde{x}_j}, \quad (\text{A.11})$$

with

$$S_{ij} = \sum_k \frac{\partial \tilde{x}_i}{\partial x_k} \frac{\partial \tilde{x}_j}{\partial x_k}.$$

For the Laplacian of the velocity, we have

$$\Delta \mathbf{v} = \sum_j \mathbf{e}_j \sum_i \frac{\partial^2 v_j}{\partial x_i^2} = \sum_i \tilde{\mathbf{e}}_i \sum_j \sum_k \frac{\partial \tilde{x}_i}{\partial x_j} \tilde{\Delta} \left(\tilde{v}_k \frac{\partial x_j}{\partial \tilde{x}_k} \right), \quad (\text{A.12})$$

where

$$\tilde{\Delta} = \sum_j \frac{\partial}{\partial x_j} \left(\sum_i \frac{\partial \tilde{x}_i}{\partial x_j} \frac{\partial}{\partial \tilde{x}_i} \right) = \sum_i \left(\sum_j \frac{\partial^2 \tilde{x}_i}{\partial x_j^2} \frac{\partial}{\partial \tilde{x}_i} + \sum_k S_{ik} \frac{\partial^2}{\partial \tilde{x}_k \partial \tilde{x}_i} \right).$$

A.2 Cylindrical Coordinates

In this section the general formulas of the previous section are going to be applied to the case where the new coordinates are cylindrical coordinates (r, θ, z) : $d\tilde{x}_1 = dr$, $d\tilde{x}_2 = r d\theta$ and $d\tilde{x}_3 = dz$. The transformation from Cartesian to cylindrical coordinates is described by

$$A_{i,j} := \frac{\partial \tilde{x}_i}{\partial x_j} \quad \mathbf{A} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (\text{A.13})$$

$$B_{i,j} := \frac{\partial x_i}{\partial \tilde{x}_j} \quad \mathbf{B} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{A.14})$$

and

$$\frac{\partial^2 \tilde{x}_i}{\partial x_j^2} = \begin{pmatrix} \frac{\sin^2\theta}{r} & \frac{\cos^2\theta}{r} & 0 \\ \frac{\sin\theta \cos\theta}{r} & -\frac{\sin\theta \cos\theta}{r} & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (\text{A.15})$$

Note that

$$\sum_j \frac{\partial \tilde{x}_i}{\partial x_j} \frac{\partial \tilde{x}_k}{\partial x_j} = \mathbf{A} \mathbf{A}^T = \mathbf{I}.$$

The unit vectors $\mathbf{e}_r, \mathbf{e}_\theta, \mathbf{e}_z$ are such that

$$\frac{\partial \mathbf{e}_r}{\partial \theta} = \mathbf{e}_\theta, \quad \frac{\partial \mathbf{e}_\theta}{\partial \theta} = -\mathbf{e}_r. \quad (\text{A.16})$$

Using Eq. (A.9) the continuity equation in cylindrical coordinates reads

$$\frac{\partial v_r}{\partial r} + \frac{v_r}{r} + \frac{1}{r} \frac{\partial v_\theta}{\partial \theta} + \frac{\partial v_z}{\partial z} = 0$$

Appendix A. Coordinate Transformations of the Navier-Stokes Equations

and using Eq. (A.11) the pressure gradient reads

$$\nabla p = \frac{\partial p}{\partial r} \mathbf{e}_r + \frac{1}{r} \frac{\partial p}{\partial \theta} \mathbf{e}_\theta + \frac{\partial p}{\partial z} \mathbf{e}_z.$$

Equation (A.10) yields that the convective term in cylindrical coordinates reads

$$\mathbf{v} \cdot \nabla \mathbf{v} = \sum_i \tilde{\mathbf{e}}_i \sum_j \tilde{v}_j \frac{\partial \tilde{v}_i}{\partial \tilde{x}_j} + \frac{v_r v_\theta}{r} \mathbf{e}_\theta - \frac{v_\theta^2}{r} \mathbf{e}_r.$$

Finally for the Laplacian of the velocity, Eq. (A.12) yields

$$\Delta \mathbf{v} = \left(\Delta_c v_r - \frac{v_r}{r^2} - \frac{2}{r^2} \frac{\partial v_\theta}{\partial \theta} \right) \mathbf{e}_r + \left(\Delta_c v_\theta - \frac{v_\theta}{r^2} + \frac{2}{r^2} \frac{\partial v_r}{\partial \theta} \right) \mathbf{e}_\theta + \Delta_c v_z \mathbf{e}_z,$$

where

$$\Delta_c = \frac{1}{r} \frac{\partial}{\partial r} + \frac{\partial^2}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2} + \frac{\partial^2}{\partial z^2}.$$

Bibliography

- [1] **Anderson D.M.; McFadden, G.B.; Wheeler, A.A.** *Diffuse-Interface Methods in Fluid Mechanics*. Annu. Rev. Fluid Mech. **30**: 139-65, 1998.
- [2] **Anjos, G.R.** *A 3D ALE Finite Element Method for Two-Phase Flows with Phase Change*, Ph.D. thesis, EPFL, 2012.
- [3] **Anjos, G.R.; Borhani, N.; Mangiavacchi, N.; Thome J.R.** *A 3D moving mesh Finite Element Method for two-phase flows*. Journal of Computational Physics **270**: 366-377 (2014).
- [4] **Arnold, D.N.; Brezzi, F.; Fortin M.** *A stable finite element for the Stokes equations.*, Calcolo **21**, 337-344, 1984.
- [5] **Bachelor, G.K.** *An Introduction to Fluid Dynamics*. Cambridge University Press (2001).
- [6] **Bach, P; Hassager, O.** *An algorithm for the use of the Lagrangian specification in Newtonian fluid mechanics and applications to free-surface flow*. J. Fluid Mech. **152**: 173-190, 1985.
- [7] **Badalassi, V.E.; Cenicerros, H.D.; Banerjee, S.** *Computation of multiphase systems with phase field models*. Journal of Computational Physics **190**: 371-397, 2003.
- [8] **Baines, M.J.; Hubbard, M.E.; Jimack, P.K.** *Velocity-Based Moving Mesh Methods for Nonlinear Partial Differential Equations* Commun. Comput. Phys. **10**(3): 509-576, 2011.
- [9] **Baines, M.J.; Hubbard, M.E.; Jimack, P.K.; Mahmood, R.** *A moving-mesh finite element method and its application to the numerical solution of phase-change problems*. Commun. Comput. Phys. **6**(3): 595-624, 2009.
- [10] **Baines, M.J.; Hubbard, M.E.; Jimack, P.K.** *A moving mesh finite element algorithm for the adaptive solution of time-dependent partial differential equations with moving boundaries*. Applied Numerical Mathematics **54**(3-4): 450-469, 2005.
- [11] **Balcazar, N.; Lehmkuhl, O.; Jofre, L.; Rigola, J.; Oliva, A.** *A coupled volume-of-fluid/level-set method for simulation of two-phase flows on unstructured meshes*. Computers and Fluids **124**, 12-29, 2016.

Bibliography

- [12] **Belhachmi, Z.; Bernardi, C.; Deparis S.** *Weighted Clement operator and application to the finite element discretization of the axisymmetric Stokes problem.* Numerische Mathematik **105**, 217-247, 2006.
- [13] **Betz, A.R.; Xu, J.; Qiu, H.; Attinger, D.** *Do surfaces with mixed hydrophilic and hydrophobic areas enhance pool boiling?* Applied Physics Letters **97**: 141909, 2010.
- [14] **Bonnerot, R.; Jamet, P.** *Numerical Computation of the Free Boundary for the Two-Dimensional Stefan Problem by Space-Time Finite Elements.* Journal of Computational Physics **25**: 163-181, 1977.
- [15] **Bernardi, C.; Dauge, M.; Maday, Y.; Azaïez, M.** *Spectral methods for axisymmetric domains.* Series in applied mathematics, Gauthier-Villars, 1999.
- [16] **Blake, T.D.; De Coninck, J.; D'Ortona, U.** *Models of Wetting: Immiscible Lattice Boltzmann Automata versus Molecular Kinetic Theory.* Langmuir **11**: 4588-4592, 1995.
- [17] **Brackbill, J.U.** *Particle Methods.* International Journal for Numerical Methods in Fluids **47**: 693-705, 2005.
- [18] **Brackbill, J.U.; Kothe, D.B.; Zemach C.** *A Continuum Method for Modeling Surface Tension.* Journal of Computational Physics **100**: 335-354 (1992).
- [19] **Braess, H.; Wriggers, P.** *Arbitrary Lagrangian Eulerian finite element analysis of free surface flow.* Comput. Methods Appl. Mech. Engrg. **190**: 95-109, 2000.
- [20] **Bretherton, F.P.** *The motion of long bubbles in tubes.* Journal of Fluid Mechanics **10**(2): 166-188, 1961.
- [21] **Brezzi, F.; Bristeau, M.O.; Franca, L.P.M; Mallet, M.; Roge, G.** *A relationship between stabilized finite element methods and the Galerkin method with bubble functions.* Computer Methods in Applied Mechanics and Engineering **96**: 117-129, 1992.
- [22] **Brooks, A.N.; Hughes, T.J.R.** *Streamline upwind/Petrov-Galerkin formulations for convective dominated flows with particular emphasis on the incompressible Navier-Stokes equations.* Comput. Methods Appl. Mech. Engrg. **32**: 199-259, 1982.
- [23] **Cao, W.M; Huang, W.Z.; Russel, R.D.** *A moving-mesh method based on the geometric conservation law.* SIAM J. Sci. Comput. **24**(1): 118-142, 2002.
- [24] **Carey, V.P.** *Liquid-vapor phase-change phenomena.* Second Edition, CRC Press, Taylor & Francis Group, 2008.
- [25] **Chen S.; Doolen, G.D.** *Lattice Boltzmann Method for Fluid Flow.* Annu. Rev. Fluid Mech. **30**: 329-364, 1998.
- [26] **Chiandussi, G.; Bugeda, G.; Onate, E.** *A simple method for automatic update of finite element meshes.* Commun. Numer. Meth. Engng. **16**, 1-19, 2000.

-
- [27] **Choi, S.; Cho, M.H.; Choi H.G.; Yoo, J.Y.** *A Q2Q1 integrated finite element method with the semi-implicit consistent CSF for solving incompressible two-phase flows with surface tension effect*. International Journal for Numerical Methods in Fluids, DOI: 10.1002/fld.4185, 2015.
- [28] **Corzo, S.F.; Damian, S.M.; Ramajo, D.; Nigro, N.M.** *Numerical Simulation of Natural Convection Phenomena*. *Mechanica Computational*, 277-296 (2011).
- [29] **Cowper, G.R.** *Gaussian quadrature formulas for triangles*. *Int. J. Numer. Meth. Engng.* **7**(3): 405-408, 1973.
- [30] **Dahmen, W.; Reusken, A.** *Numerik für Ingenieure und Naturwissenschaftler*. Springer-Verlag Berlin Heidelberg, 2008.
- [31] **Dai, M.; Schmidt, D.P.** *Adaptive tetrahedral meshing in free-surface flow*. *Journal of Computational Physics* **208**, 228-252, 2005.
- [32] **Davies, R.M.; Taylor, G.** *The Mechanics of Large Bubbles Rising through Extended Liquids and through Liquids in Tubes*. *Proc. R. Soc. Lond. A.* **200**, 375-390, 1950.
- [33] **de Gennes, P.G.; Brochard-Wyart, F.; Quere, D.** *Capillarity and Wetting Phenomena: Drops, Bubbles, Pearls, Waves*. Springer Science & Business Media, 2013.
- [34] **Deparis S.** *Numerical Analysis of Axisymmetric Flows and Methods for Fluid-Structure Interaction Arising in Blood Flow Simulation*, Ph.D. thesis, EPFL, 2004.
- [35] **Donea, J.; Huerta, A.** *Finite element methods for flow problems*. John Wiley & Sons, Ltd. (2003).
- [36] **Donea, J.; Huerta, A.; Rodriguez-Ferran, A.; Ponthot, J.-Ph.** *Arbitrary Lagrangian-Eulerian Methods*. *Encyclopedia of Computational Mechanics Vol 1* Chap. 14, John Wiley & Sons, 2004.
- [37] **Dong, Z.; Xu, J.; Jiang, F.; Liu, P.** *Numerical study of vapor bubble effect on flow and heat transfer in microchannel* *Int. J. Therm. Sci.* **54**: 22-32, 2012.
- [38] **Duprat, C.; Ruyer-Quil, C.; Giorgiutti-Dauphiné, F.** *Spatial evolution of a film flowing down a fiber*. *Physics of Fluids* **21**, 042109, 2009.
- [39] **Dziuk, G.** *An algorithm for evolutionary surfaces*. *Numer. Math.* **58**(1): 603-611, 1990.
- [40] **Eggers, J.** *Nonlinear dynamics and breakup of free-surface flows*. *Reviews of Modern Physics* **69**(3), 1997.
- [41] **Eggers, J.; Stone, H.A.** *Characteristic lengths at moving contact lines for a perfectly wetting fluid: the influence of speed on the dynamic contact angle*. *J. Fluid Mech.* **505**: 309-321, 2004.
- [42] **Eggers, J.; Villermaux, E.** *Physics of liquid jets*. *Rep. Prog. Phys.* **71** 036601, 2008.

Bibliography

- [43] **Esmaeeli, A.; Tryggvason, G.** *Computations of Explosive Boiling in Microgravity*. Journal of Scientific Computing **19**: 1-3, 2003.
- [44] **Esmaeeli, A.; Tryggvason, G.** *Computations of film boiling. Part I: numerical method*. International Journal of Heat and Mass Transfer **47**: 5451-5461, 2004.
- [45] **Esmaeeli, A.; Tryggvason, G.** *A front tracking method for computations of boiling in complex geometries*. International Journal of Multiphase Flow **30**: 1037-1050, 2004.
- [46] **Farell, P.E.; Maddison, J.R.** *Conservative interpolation between volume meshes by local Galerkin projection*. Comput. Methods Appl. Mech. Engrg. **200**: 89-100, 2011.
- [47] **Fuster, D.; Agbaglah, G.; Josserand, C.; Popinet, S.; Zaleski, S.** *Numerical simulation of droplets, bubbles and waves: state of the art*. Fluid Dyn. Res. **41** 065001, 2009.
- [48] **Ganesan, S.; Matthies, G.; Tobiska, L.** *On spurious velocities in incompressible flow problems with interfaces*. Comput. Methods Appl. Mech. Engrg. **196**: 1193-1202, 2007.
- [49] **Ganesan, S.; Tobiska, L.** *An accurate finite element scheme with moving meshes for computing 3D-axisymmetric interface flows*. Int. J. Numer. Meth. Fluids **57**: 119-138, 2008.
- [50] **Ganesan, S.; Tobiska, L.** *Modelling and simulation of moving contact line problems with wetting effects*. Comput. Visual. Sci. **12**: 329-336, 2009.
- [51] **Ganesan, S.** *Finite element methods on moving meshes for free surface and interface flows*, Ph.D. thesis, Otto-von-Guericke-University Magdeburg, 2006.
- [52] **Geuzaine, C.; Remacle, J.F.** *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*. Int. J. Numer. Meth. Engng. **79** (11): 1309-1331, 2009.
- [53] **Ginzburg, I.; Wittum, G.** *Two-Phase Flows on Interface Refined Grids Modeled with VOF, Staggered Finite Volumes, and Spline Interpolants*. J. Comput. Phys. **166**, 302-335, 2001.
- [54] **Gomez, H.; Calo, V.M.; Bazilevs, Y.; Hughes, T.J.R.** *Isogeometric analysis of the Cahn-Hilliard phase-field model*. Comput. Methods Appl. Mech. Engrg. **197** (49-50), 4333-4352, 2008.
- [55] **Gong, S.; Cheng P.** *Numerical investigation of droplet motion and coalescence by an improved lattice Boltzmann model for phase transitions and multiphase flows*. Computers and Fluids **53**: 93-104, 2012.
- [56] **Gros, E.; Anjos, G.R.; Thome J.R.** *Interface fitted moving mesh method for axisymmetric two-phase flow in microchannels*. International Journal for Numerical Methods in Fluids **86**(3), 201-217, 2017.

-
- [57] **Gross, S.** *Numerical methods for three-dimensional incompressible two-phase flow problems.*, Ph.D. thesis, RWTH Aachen, 2008.
- [58] **Gross, S.; Reusken, A.** *Numerical Methods for Two-phase Incompressible Flows.*, Springer-Verlag Berlin Heidelberg, 2011.
- [59] **Gustensen, A.K.; Rothman D.H.; Zaleski, S.; Zanetti, G.** *Lattice Boltzmann model of immiscible fluids.* Physical Review A **43**(8): 4320-4327, 1991.
- [60] **Hardt, S.; Wondra, F.** *Evaporation model for interfacial flows based on a continuum-field representation of the source terms.* Journal of Computational Physics **227**, 5871-5895, 2008.
- [61] **Harlow, F.H.; Welch, J.E.** *Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface.* Physics of Fluids **8**, 2182, 1965.
- [62] **Hirt, C.W.; Amsden, A.A.; Cook J.L.** *An arbitrary Lagrangian Eulerian computing method for all flow speeds.* J. Comput. Phys. **14**, 227-253, 1974.
- [63] **Hirt, C.W.; Nichols, B.D.** *Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries.* J. Comput. Phys. **39**, 201-225, 1981.
- [64] **Hood, P.; Taylor, G.** *Navier-Stokes eq. using mixed interpolation.* Finite element in flow problem, Oden ed. UAH Press, 1974.
- [65] **Hsieh, D.Y.** *Interfacial stability with mass and heat transfer.* Physics of Fluids **21**, 745-748, 1978.
- [66] **Huh, C.; Scriven, L.E.** *Hydrodynamic model of steady movement of a solid/liquid/fluid contact line.* Journal of Colloid and Interface Science **35**: 85-101 (1971).
- [67] **Hughes, T.J.R.; Franca, L.P.; Balestra, M.** *A New Finite Element Formulation for Computational Fluid Dynamics: V. Circumventing the Babuska-Brezzi Condition: a Stable Petrov-Galerkin Formulation of the Stokes Problem accomodating Equal-Order Interpolations.* Comput. Methods Appl. Mech. Engrg. **59**: 85-99, 1986.
- [68] **Hughes, T.J.R.; Franca, L.P.; Hulbert, G.M.** *A New Finite Element Formulation for Computational Fluid Dynamics: VIII. The Galerkin/Least-Squares Method for Advective-Diffusive Equations.* Comput. Methods Appl. Mech. Engrg. **73**: 173-189, 1989.
- [69] **Hughes, T.J.R.; W.K. Liu; Zimmermann, T.K.** *Lagrangian-Eulerian finite element formulation for incompressible viscous flows.* Comput. Methods Appl. Mech. Engrg. **29**: 329-349, 1981.
- [70] **Hysing, S.; Turek, S.; Kuzmin, D.; Parolini, N.; Burman, E.; Ganesan, S.; Tobiska, L.** *Quantitative benchmark computations of two-dimensional bubble dynamics.* Int. J. Numer. Meth. Fluids **60**, 1259-1288, 2009.

Bibliography

- [71] **Ishii, M.** *Thermo-fluid Dynamic Theory of Two-phase Flow*. Eyrolles, Paris, 1975.
- [72] **Jafari, R.; Okutucu-Oezyurt, T.** 3D numerical modeling of boiling in a microchannel by arbitrary Lagrangian-Eulerian (ALE) method *Applied Mathematics and Computation*, **272**, 593-603, 2016.
- [73] **Jafari, R.; Okutucu-Oezyurt, T.** Phase-Field Modeling of Vapor Bubble Growth in a Microchannel *The Journal of Computational Multiphase Flows*, **7**, 143-158, 2015.
- [74] **Johnson, A.A.; Tezduyar, T.E.** Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Comput. Methods Appl. Mech. Engrg.* **119**, 73-94, 1994.
- [75] **Juric, D.; Tryggvason, G.** Computations of Boiling Flows. *Int. Journal of Multiphase Flow* **24**, 387-410, 1998.
- [76] **Karayiannis, T.G.; Mahmoud, M.M.** Flow boiling in microchannels: Fundamentals and applications. *Applied Thermal Engineering* **115**: 1372-1397, 2017.
- [77] **Kawahara, M.; Anju, A.** Lagrangian finite element method for solitary wave propagation. *Computational Mechanics* **3**: 299-307, 1988.
- [78] **Kharangate, C.R.; Mudawar, I.** Review of computational studies on boiling and condensation. *International Journal of Heat and Mass Transfer* **108**: 1164-1196, 2017.
- [79] **Khodaparast, S.; Magnini, M.; Borhani, N.; Thome, J.R.** Dynamics of isolated confined air bubbles in liquid flows through circular microchannels: an experimental and numerical study. *Microfluidics and Nanofluidics* **19**, 209-234, 2015.
- [80] **Koo, J.; Kleinstreuer, C.** Viscous dissipation effects in microtubes and microchannels. *International Journal of Heat and Mass Transfer* **47**: 3159-3169, 2004.
- [81] **Lamb, H.** *Hydrodynamics*. Fourth Edition, Cambridge University Press, 1916.
- [82] **Layzer, D.** On the Instability of Superposed Fluids in a Gravitational Field. *Astrophysical Journal* **122**, 1955.
- [83] **Lee, T.E.; Baines, M.J.; Langdon, S.; Tindall, M.J.** A moving mesh approach for modelling avascular tumour growth. *Applied Numerical Mathematics* **72**: 99-114, 2013.
- [84] **Lopez, J.; Hernandez, J.; Gomez, P.; Faura, F.** An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows *J. Comput. Phys.*, **208**, 51-74, 2005.
- [85] **Liu, J.; Gomez, H.; Evans, J.A.; Hughes, T.J.R.; Landis, C.M.** Functional entropy variables: A new methodology for deriving thermodynamically consistent algorithms for complex fluids, with particular reference to the isothermal Navier-Stokes-Korteweg equations. *Journal of Computational Physics* **248**: 47-86, 2013.

-
- [86] **Lynch, D.R.** *Unified approach to simulations on deforming elements with application to phase change problems.* Journal of Computational Physics **47**: 387-411, 1982.
- [87] **Martin, D.; Chaouki, H.; Robert, J.L.; Fafard, M.** *Modelling of phase change with non-constant density using XFEM and a lagrange multiplier.* Frontiers in Heat and Mass Transfer **7**, 40, 2016.
- [88] **Masud, A.; Hughes, T.J.R.** *A space-time Galerkin/least-squares finite element formulation of the Navier-Stokes equations for moving domain problems.* Comput. Methods Appl. Mech. Engrg. **146**, 91-126, 1997.
- [89] **Monaghan, J.J.** *Smoothed particle hydrodynamics.* Annual review of astronomy and astrophysics **30**: 543-574, 1992.
- [90] **Morini, G.L.** *Viscous heating in liquid flows in micro-channels.* International Journal of Heat and Mass Transfer **48**: 3637-3647, 2005.
- [91] **Mudawar, I.** *Recent Advances in High-Flux, Two-Phase Thermal Management.* J. Thermal Sci. Eng. Appl **5**(2), 021012, 2013.
- [92] **Murea, C.M.** *Arbitrary Lagrangian Eulerian approximation with remeshing for Navier-Stokes equations.* Int. J. Numer. Meth. Biomed. Engng. **26**, 1435-1448, 2010.
- [93] **Navier, C.L.** *Mémoire sur les lois du mouvement des fluides.* Mémoires de l'Académie Royale des Sciences de l'Institut de France **6**: 389-440, 1823.
- [94] **Ong, C.L.** *Macro-to-Microchannel Transition in Two-Phase Flow and Evaporation.* Ph.D. thesis, EPFL, 2010.
- [95] **Partridge, D.** *Numerical Modelling of Glaciers: Moving Meshes and Data Assimilation.* Ph.D. thesis, University of Reading, 2013.
- [96] **Pasandideh-Fard, M.; Qiao, Y.M.; Chandra, S.; Mostaghimi, J.** *Capillary effects during droplet impact on a solid surface.* Physics of Fluids, **8**: 650, 1996.
- [97] **Picknett, R.G.; Bexon, R.** *The evaporation of sessile or pendant drops in still air.* Journal of Colloid and Interface Science **61**(2): 336-350, 1977.
- [98] **Pironneau, O.** *On the Transport-Diffusion Algorithm and Its Applications to the Navier-Stokes Equations.* Numerische Mathematik **38**, 309-332, 1982.
- [99] **Popinet, S.** *An accurate adaptive solver for surface-tension-driven interfacial flows.* J. Comput. Phys. **228**, 5838-5866, 2009.
- [100] **Popinet, S.; Zaleski, S.** *A front-tracking algorithm for accurate representation of surface tension* Int. J. Numer. Meth. Fluids, **30**, 775-793, 1999.
- [101] **Pozrikidis, C.** *Introduction to Theoretical and Computational Fluid Dynamis.* Oxford University Press, second edition, 2011.

Bibliography

- [102] **Preziosi, L.; Chen, K.; Joseph, D.** *Lubricated pipelining: Stability of core—annular flow.* Journal of Fluid Mechanics **201**: 323-356, 1989.
- [103] **Prosperetti, A.** *Free oscillations of drops and bubbles: the initial-value problem.* J. Fluid Mech. **100**: 333-347, 1980.
- [104] **Prosperetti, A.** *Boundary conditions at a liquid-vapor interface.* Meccanica **14**, 34-47, 1979.
- [105] **Prosperetti, A.; Plesset, M.S.** *The stability of an evaporating liquid surface.* The Physics of Fluids, **27**, 1590-1602, 1984.
- [106] **Quan, S.; Schmidt, D.P.** *A moving mesh interface tracking method for 3D incompressible two-phase flow.* Journal of Computational Physics **221**, 761-780, 2007.
- [107] **Quan, S.; Lou, J.; Schmidt, D.P.** *Modelling merging and breakup in the moving mesh interface tracking method for multiphase flow simulations.* Journal of Computational Physics **228**, 2660-2675, 2009.
- [108] **Quarteroni, A.; Valli, A.** *Numerical Approximation of Partial Differential Equations.* Springer Series in Computational Mathematics, Springer-Verlag Berlin Heidelberg 2008.
- [109] **Radovitzky, R.; Ortiz M.** *Lagrangian Finite Element Analysis of Newtonian Fluid Flow.* International Journal for Numerical Methods in Engineering **43**: 607-619, 1998.
- [110] **Renardy, Y.; Renardy, M.** *PROST—A parabolic reconstruction of surface tension for the volume-of-fluid method.* Journal of Computational Physics **183**, 400-421, 2002.
- [111] **Revellin, R.; Thome, J.R.** *Experimental investigation of R-134a and R-245fa two-phase flow in microchannels for different flow conditions.* International Journal of Heat and Fluid Flow **28**: 63-71, 2005.
- [112] **Roos, F.W.; Willmarth, W.W.** *Some Experimental Results on Sphere and Disk Drag.* AIAA Journal, **9**: 285-291, 1971.
- [113] **Rueda Villegas, L.; Alis, R.; Lepillez, M.; Tanguy, S.** *A Ghost Fluid/Level Set Method for boiling flows and liquid evaporation: Application to the Leidenfrost effect.* Journal of Computational Physics **316**: 789-813, 2016.
- [114] **Sato, Y.; Niceno, B.** *A sharp-interface phase change model for a mass-conservative interface tracking method.* Journal of Computational Physics **249**: 127-161, 2013.
- [115] **Sauerland, H.** *An XFEM Based Sharp Interface Approach for Two-Phase and Free-Surface Flows.* Ph.D. thesis, RWTH Aachen, 2013.
- [116] **Sauerland, H.; Fries, T.P.** *The extended finiteelement method for two-phase and free-surface flows: A system-atic study.* Journal of Computational Physics **230**: 3369-3390, 2011.

-
- [117] **Schlichting, H.** *Boundary Layer Theory*. McGraw-Hill, New York, 7-th edition 1979.
- [118] **Schlottke, J.; Weigand, B.** *Direct numerical simulation of evaporating droplets*. Journal of Computational Physics, **227**, 5215-5237, 2008.
- [119] **Schrage, R.W.** *A Theoretical Study of Interphase Mass Transfer*. Columbia University Press, New-York, 1953.
- [120] **Scriven, L.E.** *On the dynamics of phase growth*. Chemical Engineering Science **10**: 1-13 (1959).
- [121] **Shan, X.; Chen, H.** *Lattice Boltzmann model for simulating flows with multiple phases and components*. Physical Review E **47**(3): 1815-1819, 1993.
- [122] **Shewchuck, J.R.** *Triangle: engineering a 2D quality mesh generator and Delaunay triangulator*. in Applied Computational Geometry: Towards Geometric Engineering, Lecture Notes in Computer Science **1148**, 203-222, 1996.
- [123] **Shikhmurzaev, Y.** *Capillary flows with forming interfaces*. Chapman & Hall/CRC, 2008.
- [124] **Shikhmurzaev, Y.** *Singularities at the moving contact line. Mathematical, physical and computational aspects*. Physica D **217**: 121-133 (2006).
- [125] **Singh, R.; Shyy, W.** *Three-dimensional adaptive Cartesian grid method with conservative interface restructuring and reconstruction*. Journal of Computational Physics **224**(1): 150-167, 2007.
- [126] **Son, G.; Dhir, V.K.** *Numerical simulation of film boiling near critical pressures with a level set method*. J. Heat Transfer **120**: 183-192, 1998.
- [127] **Staniforth, A.; Côté, J.** *Semi-Lagrangian Integration Schemes for Atmospheric Models—A Review*. Monthly Weather Review **119**(90): 2206-2223, 1991.
- [128] **Stavrev, A.; Knechtges, P.; Elgeti, S.; Huerta, A.** *Space-time NURBS-enhanced finite elements for free-surface flows in 2D*, Int. J. Numer. Meth. Fluids, doi:10.1002/fld.4189, 2015.
- [129] **Stokes, G.G.** *On the Effect of the Internal Friction of Fluids on the Motion of Pendulums*. Transactions of the Cambridge Philosophical Society 1851, Reprinted in Mathematical and Physical Papers **3** (1880-1905).
- [130] **Sussman, M.; Smereka, P.; Osher, S.** *A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow*. J. Comput. Phys. **114**, 146-159, 1994.
- [131] **Tabata, M.; Itakura, K.** *A precise computation of drag coefficients of a sphere*. Int. J. Comput. Fluid Dynam., **9**: 303-311, 1998.
- [132] **Tai, C.F.; Chung J.N.** *A direct numerical simulation of axisymmetric cryogenic two-phase flows in a pipe with phase change*. Computers and Fluids **48**, 163-182, 2011.

Bibliography

- [133] **Tanasawa, I.** *Advances in condensation heat transfer* in: J.P. Harnett, T.F. Irvine (Eds.), *Advances in Heat Transfer*, Academic Press, San Diego, 1991.
- [134] **Tezduyar, T.E.; Behr, M.; Mittal, S.; Johnson, A.A.** *Computation of unsteady incompressible flows with the finite element methods - space-time formulations, iterative strategies and massively parallel implementations.* in P. SMolinski, W.K. Liu, G. Hulbert, and K. Tamma, editors, *New Methods in Transient Analysis*, AMD **143**, ASME, 7-24, 1992.
- [135] **Tezduyar, T.E.; Behr, M.; Liou, J.** *A new strategy for finite element computations involving moving boundaries and interfaces - the deforming-spatial-domain/space-time procedure: I. The concept and the preliminary tests.* *Comput. Methods Appl. Mech. Engrg.* **94**, 339-351, 1992.
- [136] **Thomas, P.D.; Lombard, C.K.** *The geometric conservation law and its application to flow computations on moving grids.* *AIAA J.* **17**: 1030-1037, 1979.
- [137] **Thome J.R.** *Boiling in microchannels: a review of experiment and theory.* *Int. J. Heat and Fluid Flow* **25**: 128-139 (2004).
- [138] **Tomar, G.; Biswas, G.; Sharma, A.; Agrawal, A.** *Numerical simulation of bubble growth in film boiling using a coupled level-set and volume-of-fluid method.* *Physics of Fluids* **17**: 112103, 2005.
- [139] **Tryggvason, G.; Bunner, B.; Esmaeeli, A.; Juric, D.; Al-Rawahi, N.** *A Front-Tracking Method for the Computations of Multiphase Flow.* *Journal of Computational Physics* **169**, 708-759, 2001.
- [140] **Tryggvason, G.; Esmaeeli, A.; Al-Rawahi, N.** *Direct numerical simulations of flows with phase-change.* *Computers and Structures* **83**, 445-453, 2005.
- [141] **Waldrop, M.M.** *More than Moore.* *Nature* **530**, 144-147, 2016.
- [142] **Walkley, M.A.; Gaskell, P.H.; Jimack P.K.; Kelmanson, M.A.; Summers, J.L.** *Finite element simulation of three-dimensional free-surface flow problems with dynamic contact lines.* *International Journal for Numerical Methods in Fluids* **47**: 1353-1359, 2005.
- [143] **Welch, S.W.J.** *Local simulations of two-phase flows including interface tracking with mass transfer.* *Journal of Computational Physics* **121**, 142-154, 1995.
- [144] **Welch, S.W.J.** *Direct simulation of vapor bubble growth.* *International Journal of Heat and Mass Transfer* **41**, 1655-1666, 1998.
- [145] **Welch, S.W.J.; Wilson, J.** *A Volume of Fluid based Method for Fluid Flow with Phase Change.* *Journal of Computational Physics* **160**, 662-682, 2000.
- [146] **Wick, T.** *Fluid-structure interactions using different mesh motion techniques.* *Computers and Structures* **89**: 1456-1467, 2011.

- [147] **Yan, H.; Wang, Z.; He, J.; Chen, X.; Wang, C.; Peng, Q.** *Real-time fluid simulation with adaptive SPH* Comp. Anim. Virtual Worlds **20**: 417-426, 2009.
- [148] **Ye, T.; Shyy, W.; Tai, C.F.; Chung J.N.** *Assessment of sharp- and continuous-interface methods for drop in static equilibrium*. Computers and Fluids **33**(7): 917-926, 2004.
- [149] **Zienkiewicz, O.C.; Taylor, R.; Nithiarasu, P.** *The finite element method for fluid dynamics (6th Edition)*. Butterworth-Heinemann, 2005.

Curriculum Vitae - Erik Gros

Education:

March 2015 - June 2018: PhD in Mechanical Engineering, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

February 2013 - October 2014: Master in Aerospace Engineering, RWTH Aachen University, Aachen, Germany

October 2008 - February 2013: Bachelor in Mechanical Engineering, RWTH Aachen University, Aachen, Germany

September 2004 - June 2008: High School, Athénée Royal de Watermael-Boitsfort, Brussels, Belgium

Work Experience:

March 2015 - June 2018, EPFL, Heat and Mass Transfer Laboratory (LTCM)

Job description: research and teaching assistantship

October 2013 - April 2014, RWTH Aachen University, Chair for Computational Analysis of Technical Systems (CATS):

Job description: undergraduate research about development of a finite element flow solver for the simulation of viscoelastic flow

March 2012 - September 2012, Eurocopter Deutschland GmbH, Donauwörth, Germany:

Job description: project management internship about the production of helicopter and airplane door parts

Teaching Experience:

École Polytechnique Fédérale de Lausanne (EPFL):

- Heat and Mass Transfer (Spring 2018)
- Two Phase Flow and Heat Transfer (Fall 2016, Fall 2017)
- 2nd Workshop on Advances in Theoretical and Computational modeling of Interface Dynamics in Capillary Two-Phase Flows (October 9-13, 2016)
- Numerical Methods in Heat Transfer (Spring 2017)

RWTH Aachen University:

- Fluid Dynamics (WS 2011/12)
- Solid Mechanics (WS 2010/11, SS 2011, WS 2011/12)
- Simulation Technology (SS 2011)

Additional Training and Courses:

- Methods of Asymptotic Analysis in Mechanics, Prof. P. Monkewitz, EPFL (Spring 2017)
- 1st Workshop on Advances in CFD and MD modeling of Interface Dynamics in Capillary Two-Phase Flows, EPFL (October 3-7, 2016)
- Fourier Analysis and Boundary Value Problems, Prof. A. Rinaldo, EPFL (Spring 2016)
- Iterative Solvers and Parallelization, High-Performance Computing Center Stuttgart (March 16-20, 2015)
- Parallel Programming with MPI and OpenMP, Jülich Supercomputing Center (December 1-3, 2014)
- Lattice Boltzmann Methods, Prof. G. May, RWTH Aachen University (WS 2014/15)

Computer Skills:

Operating Systems: UNIX (OSX, Ubuntu), Windows.

Scientific Computing: C++, Dymola, FORTRAN, git, MATLAB, OpenMP, Paraview, Python.

Writing/Presentations: LaTeX, Microsoft Word and PowerPoint.

Languages:

English (professional proficiency), French (native), German (fluent) and Italian (fluent).

Conference Talks and Seminars:

- E. Gros, G. Anjos, J. Thome; *An ALE FEM method for two-phase flow with phase change*; 53rd European Two-Phase Flow Group Meeting; May 22-24, 2017, Gdansk (Poland)
- E. Gros, G. Anjos, J. Thome; *An ALE Finite Element Approach for Two-Phase Flow with Phase Change*; 69th Annual Meeting of the APS Division of Fluid Dynamics; November 20-22, 2016, Portland (USA)
- E. Gros; *Interface fitted moving mesh method for two-phase flow simulation*; Capillary and Two-Phase Flows IGM Seminar; June 20, 2016, Lausanne (Switzerland)
- E. Gros; *Numerical Modelling of Two-Phase Flows with Moving Contact Lines*; PPG-EM Seminar; August 19, 2015, Rio de Janeiro (Brazil)

Published work:

- E. Gros, G. Anjos, J. Thome; *Arbitrary Lagrangian Euler Method for Two-Phase Flows: 2D and Axisymmetric Formulation*; Encyclopedia of Two-Phase Heat Transfer and Flow III, Vol 1, Chap 6; 2018 (book chapter)
- E. Gros, G. Anjos, J. Thome; *Interface fitted moving mesh method for axisymmetric two-phase flow in microchannels*; International Journal for Numerical Methods in Fluids 86(3): 201-217; 2017 (journal article)
- E. Gros, G. Anjos, J. Thome; *An ALE-FEM Method for Axisymmetric Two-Phase Flows*; Proceedings of COBEM 2017, 24th ABCM International Congress of Mechanical Engineering; December 3-8, 2017 (conference paper)
- G. Anjos, E. Gros, J. Peixinho; *Rising Bubble in a vertical slowly diverging Pipe*; 9th World Conference on Experimental Heat Transfer, Fluid Mechanics and Thermodynamics; Iguazu (Brazil); June 12-15, 2017 (conference paper)

