COPPE
UFRJ
Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia

FINITE ELEMENT SIMULATION OF THREE-DIMENSIONAL TWO-PHASE FLOWS IN COMPLEX GEOMETRIES

Daniel Barbedo Vasconcelos Santos

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Mecânica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Mecânica.

Orientador: Gustavo Rabello dos Anjos

Rio de Janeiro
Fevereiro de 2024

FINITE ELEMENT SIMULATION OF THREE-DIMENSIONAL TWO-PHASE FLOWS IN COMPLEX GEOMETRIES

Daniel Barbedo Vasconcelos Santos

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA MECÂNICA.

Orientador: Gustavo Rabello dos Anjos

Aprovada por: Prof. Gustavo Rabello dos Anjos
Prof. Alvaro Luiz Gayoso de Azeredo Coutinho
Prof. Enio Pedone Bandarra Filho
Prof. Gustavo César Rachid Bodstein
Prof. Norberto Mangiavacchi
Prof. Su Jian

RIO DE JANEIRO, RJ – BRASIL
FEVEREIRO DE 2024

Barbedo Vasconcelos Santos, Daniel

Finite Element Simulation of Three-Dimensional Two-Phase Flows in Complex Geometries/Daniel Barbedo Vasconcelos Santos. – Rio de Janeiro: UFRJ/COPPE, 2024.

XVII, 112 p.: il.; 29, 7cm.

Orientador: Gustavo Rabello dos Anjos

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Mecânica, 2024.

Referências Bibliográficas: p. 101 – 110.

1. Finite Element Method. 2. Front-Tracking. 3. Laplace-Beltrami. 4. Semi-Lagrangian. I. Rabello dos Anjos, Gustavo. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Mecânica. III. Título.

Acknowledgements

Firstly, I want to thank my advisor, Professor Gustavo Rabello. This thesis wouldn't have been possible without his knowledge, guidance, patience, and friendship. I appreciate him accepting me as a student, and I couldn't have asked for a better advisor.

I also want to express gratitude to my friends and family. Thanks to Rayssa for listening to my complaints about the semi-Lagrangian issues, and to Raquel, Carlos, Jean, Larissa, Gabriel, and Natalia for their friendship and support. Special thanks to Roberta for welcoming me to the laboratory and making me feel at home.

I extend my thanks to CAPES for their funding that supported my studies. Additionally, I would like to express my gratitude to LabMFA for providing access to the necessary computing resources. All simulations presented in this work were executed on the computers made available by LabMFA.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

FINITE ELEMENT SIMULATION OF THREE-DIMENSIONAL TWO-PHASE FLOWS IN COMPLEX GEOMETRIES

Daniel Barbedo Vasconcelos Santos

February/2024

Advisor: Gustavo Rabello dos Anjos

Department: Mechanical Engineering

In this work, three-dimensional two-phase flows in complex geometries are investigated through finite element numerical simulations. The governing equations, are written in three-dimensional Cartesian coordinates and solved through the finite element method on unstructured meshes, using the Taylor-Hood "Mini" element, chosen due to its numerical stability and convergence. The convective term of the momentum equation is discretized by a first-order semi-Lagrangian method. The fluid phases are separated by an interface mesh constructed by triangular surface elements, uncoupled from the main three-dimensional fluid mesh. Surface tension force is calculated through the Laplace-Beltrami operator and added to the simulation as a source term through the continuum surface force model (CSF). At the start of each time step, the positions of the interface mesh nodes are updated based on the fluid velocity fields. The proposed methodology is verified against analytical and experimental data through single and two-phase benchmark tests found in the literature, such as the static droplet, oscillating droplet, rising bubble in quiescent fluid and Taylor bubbles, and the numerical code is used to simulate several two-phase flows in complex geometries, such as Taylor bubbles in different cross-sections and converging, expanding and curved channels.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

Daniel Barbedo Vasconcelos Santos

Fevereiro/2024

Orientador: Gustavo Rabello dos Anjos

Programa: Engenharia Mecânica

Neste trabalho, escoamentos bifásicos tridimensionais em geometrias complexas são investigados através de simulação numérica utilizando o método dos elementos finitos. As equações governantes, são escritas em coordenads Cartesianas tridimensionais e solucionadas numericamente através do método dos elementos finitos em malhas não estruturadas, usando o elemento da família Taylor-Hood, elemento "Mini". O termo convectivo da equação da quantidade de movimento é discretizado por um esquema semi-Lagrangiano de primeira ordem. As fases fluidas são separadas por uma malha interfacial construida por elementos triangulares de superfície, separada da malha tridimensional principal do fluido. A força de tensão superficial é calculada do operador Laplace-Beltrami, e adicionada à simulação como um termo fonte através método *continuum surface force* (CSF). No início de cada iteração a posição dos nós da malha da interface é atualizada pelos campos de velocidade do fluido. A metodologia proposta é verificada por testes com uma ou duas fases encontrados na literatura, tais como a gota estática, gota oscilante, bolha ascendendo em fluido em repouso e bolhas Taylor, e o código numérico é usado para simular vários escoamentos bifásicos em geometrias complexas, como bolhas Taylor em diferentes seções transversais, e canais convergente, em expansão e curvo.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The presence of two-phase flow can be seen across a broad spectrum of industrial applications, including cooling of high-tech components such as microprocessors and data centers [1], oil and gas production and refinement [2], [3], solar and nuclear energy generation, [4], [5], medical X-ray equipment, power electronics in hybrid vehicles, and beyond [6]. The complexity of two-phase flow stems from its rich physics, encompassing heat transfer, evaporation, condensation, capillary effects, and bubble dynamics, all within a single system. Consequently, it is a complex phenomenon to model, whether experimentally or numerically.

One particular interest of the present research regarding the applications of two-phase flow is the cooling of electronic components, an interest shared by the project ACoolTPS - Advanced Cooling of high power microsystems using Two-Phase Flows Systems in complex geometries, within the Royal Society Newton Advanced Fellowship. Microprocessor chips are becoming faster and smaller, and are present in an ever-growing amount of devices, such as handhelds, consumer appliances, cars, data centers, and even in satellites and space stations. The advancements in speed and miniaturization of electronic devices have led to a significant increase in heat generation. Operating microprocessors outside of their prescribed temperature range can result in costly failures, affecting not just governments and organizations, but also individuals. The present conventional cooling approaches are becoming ineffective in removing the heat from current electronic devices as well as future ones being designed, creating a demand for better thermal management solutions.

To meet this demand, several approaches have been studied. There is research on the optimization of fin configuration and heat sink geometry [7], [8]; the use of nanofluids as coolants [9], [10], [11]; and design, improvement, and application of thermoelectric devices to cooling solutions [12], [13]. One promising alternative to current heat dissipation methods is the use of phase-change-based cooling devices. These devices are lightweight, energy efficient, and small in size, allowing them to be put directly in contact or in very close proximity to the microprocessors. They

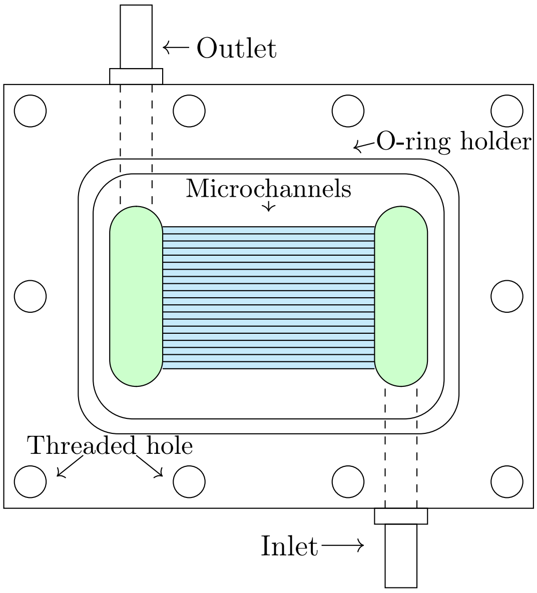


Figure 1.1: Microchannel based evaporator schematics. Fluid is pooled at the end of the inlet and outlet tubes (highlighted in green) and is conducted along the microchannels in the middle (highlighted in blue), in parallel, from right to left. The microprocessor can be positioned beneath the microchannel zone.

use a phase-change refrigerant-type coolant which changes into vapor phase when exposed to heat. The latent heat of vaporization is orders of magnitude greater than the specific-heat capacities of coolants in non-phase-change methods, improving the solution efficacy. The schematics for the evaporator of such device can be observed in fig. 1.1

However, designing phase-change devices presents a significant challenge. Two-phase flow inside the microchannels can have several different flow regimes, including bubbly flows, stratified flows, slug flows, plug flows, and misty flows. Understanding the two-phase flow mechanics is essential to achieving effective designs and one way to accomplish this is through numerical simulation.

Numerical simulation is a powerful tool to describe two-phase phenomena. It offers speed and economical advantage over experimental setups and it can depict scenarios that can be hard to observe as well as gather data in an experimental setup, such as two-phase flow in microchannels, whose characteristic length can be around microns in size.

Modelling such complex phenomena numerically is not an easy task, however. The Navier-Stokes equations display non-linearity in the material derivative term which leads to spurious currents in the results depending on simulation parameters when discretizing the equations through the classical Galerkin method. The representation and tracking of the moving interface between fluids also offers a hard challenge, where many approaches have been used over the years, each presenting advantages and downsides, without a clear best approach to the issue. The simulation might also have to deal with a sharp transition of specific mass and viscosity across the fluid interface, possibly introducing numerical instability due to the nature of the numerical operations.

This thesis proposes an enhanced methodology for three-dimensional two-phase flow simulations, extending the uncoupled formulation described in [14]. An Eulerian description and the Finite Element method are used to describe the flow field based on the "one-fluid" approach, while having a separate Lagrangian description for the fluid interface. The Navier-Stokes equations were discretized over a three-dimensional domain, using an unstructured finite element mesh of Mini tetrahedron elements. The convective non-linear terms of the Navier-Stokes equations are represented by a linear semi-Lagrangian method, which has no time step limitation and is unconditionally stable, as shown in [15]. There are many methods to delimit the fluid interface, and for this work, an unstructured three-dimensional surface mesh serves as a boundary between the two phases. This interface mesh is not coupled to the fluid mesh, but its position is updated at each time step using the velocity field data obtained from the finite element solution of the fluid flow. The surface tension, one of the main forces acting upon the fluid in a two-phase flow, is calculated using the interface mesh mean curvature. This mean curvature is obtained through a finite element discretization of the Laplace-Beltrami operator applied to the interface's surface mesh. Due to the nature of the chosen method to represent the fluid interface, some finite elements of the fluid mesh may be intersected by the interface mesh. In these elements, the fluid properties, namely specific mass and viscosity, are averaged over the element using their nodal values. This element intersection by the interface can generate numerical instability; therefore an artificial thickness is assigned to the fluid interface using a Heaviside function to attenuate the issue. Following the proposed methodology, several two-phase flows are simulated, and several complex geometry two-phase flow test cases are presented, with their results compared to existing data.

The methodology is laid over the listed chapters, followed by test cases offering validation for three-dimensional one-phase flow numerical simulation, as well as the mean curvature obtained by the Laplace-Beltrami operator. Two-Phase flow examples with known results are exhibited afterward, to demonstrate the accuracy of the proposed two-phase methodology, and lastly, several test cases composed of two-phase flows over complex geometry are presented and the results discussed. These test cases intend to simulate microchannels of diverse cross sections, including square, triangular, converging, and diverging. The content of this work is organized as follows:

- Chapter 1: Introduction

- Chapter 2: Literature Review
- Chapter 3: Methodology
- Chapter 4: Numerical Results
- Chapter 5: Conclusion
- Appendix A: Publications

Chapter 2

Literature Review

A couple of decades after its development as a tool to solve problems in solid mechanics, the Finite Element Method started being applied to the solution of fluid flow problems. It offered an interesting alternative to the finite difference method due to the ease which the boundary conditions can be set, irregular geometries with convex shapes modeled, and unstructured meshes used. It was soon applied to the solution of field problems, [16], solution to unsteady irrotational flows and led to the development of isoparametric elements [17], which can better represent curved surfaces and arbitrarily shaped boundary conditions.

The simulation of two-phase flows involves solving a challenging problem, specifically the identification of the two fluid phases and the boundary that separates them, their interface. Strictly speaking, this interface consists of a very small, thin region where molecules from both fluids can be found, with a thickness equivalent to just a few mean free molecular paths. However, in engineering applications, the scale of this interface is often significantly smaller compared to the scale of the overall problem. This effectively results in a near-zero thickness, representing a sharp boundary between the fluids with a steep gradient in fluid properties.

In general, when using a fixed grid, the identification of fluid phases can be achieved through a marker function. As fluids flow and change position, this marker function requires periodic updates. Various methods are available for this purpose, many of which are based on two different approaches: interface capturing and interface tracking methods.

2.1 Interface Capturing

The first approach, interface capturing, tracks the phases positions implicitly based on a scalar field associated with the background mesh. The interface is not explicitly presented and must be constructed using information obtained from the scalar field. This Eulerian approach is called interface capturing, and an example of it is presented in fig. 2.1c. The scalar field which delimits and identifies the fluids is also advected by velocity interpolated from the fixed fluid mesh. Bubble coalescence and break-up are modeled more easily using this approach, and complex geometrical changes are handled automatically. As disadvantages, the interface capturing methods require a more refined fixed mesh compared to interface tracking methods, if the same level of accuracy is desired. Some methods also have a less accurate representation of the interface and therefore poorer mean curvature calculation, which can lead to less precise surface tension representation.

A popular interface capturing method is the Volume of Fluid (VOF) [18]. In the VOF method applied to finite elements, each node is subjected to a function F, which assigns either a value of one if the node contains one of the phases, or zero if it contains the other. These values are averaged for each finite element, therefore the boundary between fluids, the interface, lies over elements that present a number between one and zero. The function F is then advected by the fluid velocity and the function values at the nodes are updated. The VOF function is discontinuous, and accuracy limitations when computing the curvature and interface normal, arise because of that. Extra steps are needed to recover the interface explicitly, and the results are not unique, depending on the technique developed for the task. Many works were developed based on the VOF, such as [19], where the researchers developed a hybrid Volume of Fluid-Immersed Boundary Method (VOF-IBM) method to describe numerically the freezing of liquid drops and liquid films. The equations for heat exchange and phase change were coupled, and the surface tension effects at the liquid-gas interface were taken into account, as well as the volume expansion due to the specific mass difference between water and ice. They found that the final frozen drop shape matches the experimental data, including the tip angle of the frozen drop and the front-to-interface angle. [20] used the VOF method coupled with the LES turbulence model to simulate a dam break flow in three dimensions. The authors intended to use the model to be able to identify flooding zones and flooding time, critical parameters for evacuation of people from dangerous areas. The accuracy of their proposed model was estimated by comparing the obtained results to other numerical analyses and experimental data. Their model was considered applicable to study the initial stages of a dam break, the formation of underwater vortexes, waves on the water surface, and backward waves. [21] developed a three-dimensional multi-scale Discrete Element Method (DEM) coupled with a VOF solution to solve three-phase flow configurations in a computationally efficient way. The method is useful in situations where large geometric scale quantities are of interest, but small-scale phenomena cannot be discarded. A dual mesh is used, and information from the small scale is fed into the bulk scale to reach the solution. The method produced results agreeable with the literature and was able to reproduce experiments with more accuracy than traditional DEM-VOF methods. In [22] new methodologies are proposed to address the problems that affect the VOF model, namely the difficulties in the reconstruction of the interface which lead to erroneous mean curvature calculations and the smearing of the interface due to inaccurate artificial diffusion that appears on the front. The author also tried to address a problem that afflicts several two-phase flow simulation techniques, the spurious velocities which appear on the material discontinuity at the fluid interface. Several classical two-dimensional benchmarks were presented, tank draining, rising bubble, Rayleigh-Taylor instability, and dam-break flows, and two were proposed involving the merging of two rising bubbles of different diameters and properties.

The Level Set method is another popular interface capturing method, which was introduced in [23]. It consists in embedding a function representing the interface, $\Gamma(t)$ in a function $\Phi(\mathbf{x}, t)$. $\Gamma(t)$ is a surface on a three-dimensional simulation or a curve on a two-dimensional simulation. $\Phi(\mathbf{x}, t)$ is defined in such a way that for each point in the domain of $\Gamma(t)$, $\Phi(\mathbf{x}, t) = v_{ls}$. The value v_{ls} is the level set value, which is arbitrary, but it is usually set to zero. If it is chosen as zero, the fluid phases can be identified by just the sign of $\Phi(\mathbf{x}, t)$. A convenient option for the function $\Phi(\mathbf{x}, t)$ is a distance function, where $v_{ls} = 0$. As the fluids flow, the interface is advected by the fluids' velocity fields. The Level Set method allows for ease of calculation of the interface's curvature and normal vector. The method offers a fully implicit interface, which may be reconstructed globally and is unique. A drawback of the method is that mass conservation is not guaranteed as the simulation advances in time. Examples of methods based on the Level Set are [24] which presents a two-dimensional three-phase flow numerical model with phase change. The goal is to simulate flows with two immiscible fluids and a third phase composed of one of the fluid's vapor. Two-level set functions are used to capture the two interfaces involved in the problem. The numerical solution is executed on a fixed mesh using the finite volume method. The model was found applicable for problems involving condensation or vaporization and solidification or melting, accounting for conjugate heat transfer in the latter. [25] proposes an association of the level set method with an interface conforming method that follows the interface with an adaptive unstructured mesh for solving two-phase or free surface flows. The domain evolution is dealt with implicitly, while a sharp representation of the interface by an explicit manifold is achieved. The author executed free surface and two-phase benchmarks in two and three dimensions. [26] uses a Level Set method with global mass correction for simulating free surfaces and interfaces. The incompressible two-dimensional Navier-Stokes equations are discretized by the finite volume method on a staggered grid with a two-step projection method, and the mass correction is applied to one fluid, automatically correcting the other. The results obtained are in agreement with theoretical predictions, experimental data, and previous numerical simulations. The mass errors encountered after the mass correction were close to machine zero.

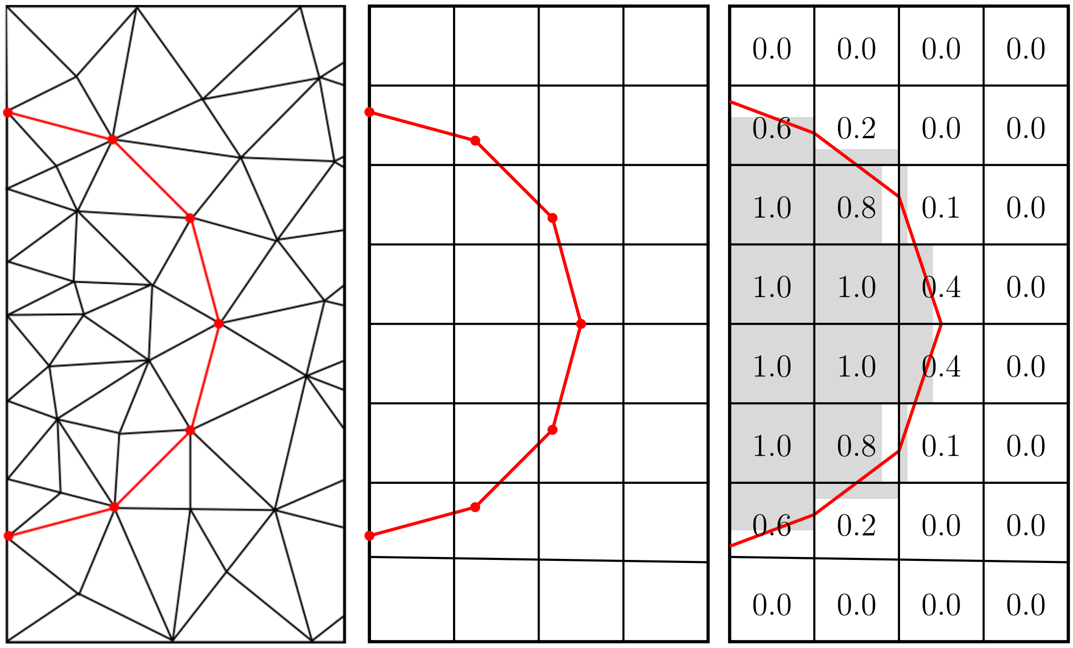
Other methods based on the interface-capturing approach were developed over time, such as the CLSVOF (Coupled Level Set Volume of Fluid), which was created in efforts to reduce the drawbacks of both the Level Set and Volume of Fluid methods [27], the Flexible Coupled Level Set and Volume of Fluid presented by [28], the VOSET method by [29] and others [30], [31].

2.2 Interface Tracking

The second approach, interface tracking, follows the interface position explicitly, either by marking a subset of interconnected nodes belonging to the fluid mesh or by introducing a secondary mesh not coupled to the fluid mesh. These mesh nodes are advected by interpolation of the velocity values obtained from the Navier-Stokes solution calculated on the fixed mesh, also referred to as the background mesh. This Lagrangian approach is called interface tracking, and it offers a sharp interface representation and accurate property jumps, especially in methods where the secondary mesh is coupled to the background fluid mesh. Examples of this approach can be seen in figs. 2.1b and 2.1a, where one can observe an uncoupled and coupled fluid interface, respectively. This avoids numerical diffusion completely at the interface and reduces the mesh refinement needed at the front. One major drawback of this approach is that the mesh nodes representing the interface tend to cluster or scatter in certain areas, causing distortion and possibly tangling. To overcome this issue, a remeshing process is required. Another issue is that these methods cannot deal with coalescence or break-up naturally, so an explicit model is necessary to simulate those situations.

In the Front Tracking method, used in this work, the interface between fluids is created explicitly from the fluid or background mesh. This interface is one dimension lower than the fluid interface and is completely separated from the fluid mesh. The interface is advected by velocity obtained from the interpolation of fluid mesh nodal values, moving through the fluid mesh as time advances. The explicit treatment of the interface keeps the interface sharp, the fluid properties well defined and allows the inclusion of surface tension forces at the interface. Between the drawbacks of the method, one can cite the added implementation complexity due to the two-way interaction of the interface with the fluid, the necessity of restructuring the interface mesh as it deforms beyond usability due to uneven fluid velocity, and the necessary explicit treatment of interface to interface contact. One popular front-tracking work was developed in [32], using the Finite Difference method, on a stationary grid. The author chose to give the interface a finite thickness of the same order as the fluid mesh element size to improve stability and smoothness, this thickness is constant over time and therefore does not cause numerical diffusion. The fluid properties, specific mass, and viscosity are updated according to an indicator function, which produces the value of 0 outside the bubble, and 1 inside, and is constructed by the known interface position. [33] developed a new numerical algorithm for the front-tracking method based on an adaptive anisotropic unstructured mesh. In the author's work, both the fluid mesh and the interface mesh are unstructured anisotropic meshes, with the goal of improving modeling capabilities of complex geometries. Special attention was given to physical variable interpolation between the meshes. This algorithm was implemented on the commercial CFD package ANSYS Fluent through user-defined functions. The author presented three test cases as validation, a gas bubble rising in quiescent liquid, a buoyancy-driven liquid droplet rising in a periodically constricted capillary tube, and a pressure-driven droplet passing through a small hole in a pipe. The results were reasonably agreeable with experimental data. [34] developed a front-tracking method to simulate the evaporation of liquid droplets, driven by temperature or species gradient, in two dimensions. The equations are solved by a finite difference method, using a Cartesian uniform grid. The authors present a species gradient phase change model, where the mass transfer occurs due to a species mass fraction gradient, even when the temperature is the same for both phases. The Clausius-Clapeyron relation is used to compute the species mass fraction and the evaporation mass flux at the interface.

The Arbitrary Lagrangian-Eulerian (ALE) formulation, which is based on front-tracking, was developed in an effort to combine the good characteristics of both the Lagrangian and Eulerian approaches. In the ALE formulation, the computational mesh nodes can be moved with the continuum. Due to this freedom, parts of special interest to the simulation can be advected, like the fluid interface in two-phase flow for example, while maintaining good quality elements and allowing greater continuum deformation than a purely Lagrangian approach would. The ALE formulation allows for the computational elements to move position, because of that, in two-phase flows, the interface can be always positioned at an element boundary. This allows for a sharp interface, with no necessity of a finite thickness, and very well-defined properties in both fluid phases. It can also offer selective refinement, improving mesh quality on selected regions as the simulation advances in time. This is particularly useful on two-phase flows, where for example one might want to refine the fluid interface or the region occupied by a bubble or droplet. One big disadvantage of the ALE formulation is that the movement of the fluid mesh eventually deforms the mesh beyond usage, through collapsing of elements and mesh degeneration. To avoid that, a re-meshing or mesh update procedure is necessary.



0.0	0.0	0.0	0.0
0.6	0.2	0.0	0.0
1.0	0.8	0.1	0.0
1.0	1.0	0.4	0.0
1.0	1.0	0.4	0.0
1.0	0.8	0.1	0.0
0.6	0.2	0.0	0.0
0.0	0.0	0.0	0.0

(a) Interface Conforming (ALE) over an unstructured mesh in two dimensions.

(b) Front-Tracking method over a structured mesh in two dimensions.

(c) Volume Tracking method over a structured mesh in two dimensions.

Figure 2.1: In a) an Interface Conforming example is presented. The mesh adapts so that the interface is represented by the edges of a subset of elements. In the picture, the element edges that separate the fluids are marked in red. In b) a Front-Tracking method is displayed. The red line represents the fluid interface, called the "front", which might be represented by line elements in two dimensions. The curvature calculations are based on the fluid interface. In c) a Volume Tracking method is exhibited. The numbers in each cell mark the percentage of one of the fluid phases inside each cell. The interface is not represented explicitly but might be re-constructed from information on the amount of fluid in each cell, using a variety of methods. Represented in the figure is the SLIC reconstruction method.

There are several works detailing ALE formulations, [35] presents a numerical method for two-phase fluid flow with heat transfer and phase change, using the finite element method combined with the ALE framework using an axisymmetric domain. The interface was followed with an isotherm moving mesh and automatic re-meshing is triggered when large mesh deformation happens. The thermodynamic equilibrium approach was used to model the mass transfer rate at the interface, which assumes the temperature of both fluids is the same at the interface, which is at saturation conditions. This temperature is considered continuous and constant over the interface. Significant accuracy for the surface tension representation was reported by the method. The accuracy of the simulation was verified through commonly used benchmarks for phase change problems. [36] proposed a finite element based method for modeling compressible multi-phase flow that allows discontinuities of some variables at the interface by using discontinuous basis functions while using C^0 finite element basis functions outside the interface. The mesh at the interface is captured by an ALE framework, moving with the interface, and being updated when the mesh quality is no longer adequate. The author's goal is to simulate high-rate phase change phenomena, such as the collapse of bubbles or the combustion of densely energetic materials. Such phenomena can lead to very high flow speeds in the lighter phase, which assumes compressible behavior with possible multiple temporal scales. The authors plan to include the addition of a surface tension term to the equations in order to simulate the collapse and interaction of water vapor bubbles. [37] presented an interface tracking ALE finite element method for the simulation of axisymmetric two-phase flows, with dynamic boundaries. In this work, the mesh nodes move to give a detailed description of the fluid interfaces while using adaptive mesh refining and remeshing to guarantee high-quality mesh elements, being useful for very large or periodic fluid domains. The method was validated by simulating the sessile test case and the rising of single bubbles and drops with different fluid properties in challenging geometry, for example, a divergent channel and a corrugate channel. The results were compared with exact results and other authors' data, and the method was considered an accurate simulation tool.

Additionally, [38] solved the incompressible Navier-Stokes equations for a two-dimensional domain by discretizing them by Taylor-Hood elements, using an ALE finite element method for an axisymmetric domain, where the mesh conforms to the fluid interface, and re-meshes when the interface suffers deformations, following its evolution. The research focus on developing an accurate and robust method for interfacial flows driven by strong surface tension and weak viscous dissipation, using an interface conforming method but eliminating spurious velocities, typically associated with interface conforming methods. The technique is validated through a series of benchmark tests with large interface deformations, in particular, bubble and droplet oscillations and dripping faucet are compared to theoretical and experimental results available in the literature. [39] built a volume-averaged ALE finite element model algorithm for predicting macrosegregation coupled with shrinkage cavity, which has a strong impact on macrosegregation, in steel ingots. Macrosegregation is a macroscopic scale composition heterogeneity, that is a serious defect in steel ingots. It forms during the solidification process, and it is caused by the relative motion of solute-depleted solid phase and the solute-rich liquid phase. The moving mesh was used to model the shrinkage cavity that forms while the material solidifies.

Other methods based on front-tracking were developed. For example, the phase field method which was first developed as a tool for modeling solid-liquid phase transitions where surface tension effects and non-equilibrium thermodynamic behavior are significant at the surface [40], [41], [42].

The finite element discretization of the non-linear advective term of the Navier-Stokes equations is a source of numerical instability when the mesh Peclet number exceeds a critical value, which is often too small to be practical. There are some strategies to deal with this instability, for example the Streamline Upwind/Petrov-Galerkin method which fixes the excessive diffusion of previous upwind schemes by adding diffusion only in the flow direction or the Least Squares Galerkin for elliptic and second-order hyperbolic problems which achieve stability through the addition of least square residuals to the Galerkin method. The method used on the present work, the semi-Lagrangian method, treats the advective term explicitly. It is advantageous because it is unconditionally stable, and preserves the linear system of equations symmetry. It has been used on many numerical applications. [43] used the semi-Lagrangian to accurately solve the convective term in a multilevel adaptive enriched finite element method for simulation coupled flow-transport problems on unstructured triangular meshes. The authors provided a number of numerical benchmarks, such as flow past a circular cylinder and heat transport problem in the Mediterranean Sea. [44] uses the semi-Lagrangian method to resolve the streaming step in a Lattice Boltzmann method variant which uses non-uniform grids, to simulate incompressible flows. The authors conducted several numerical tests to verify the method, including flow past a stationary cylinder, the lid-driven cavity test and flow past an NACA0012 airfoil, finding the results in good agreement with the literature. [45] proposed a semi-Lagrangian approach to circumvent the strict time step sizes for high-flow velocities typical of many high-speed compressible flows. Using a semi-Lagrangian Lattice Boltzmann method, the authors demonstrated stability for time steps that exceed typical CFL constraints in two and three dimensional compressible flows. [46] uses the semi-Lagrangian scheme to discretize the advection part of the Navier-Stokes equations on his semi-implicit non-linear volume of fluid approach to simulate complex free-surface flows, using a combination of a finite volume scheme for the discretization of the hydraulic head and finite differences for the momentum equations.

The novelty of this thesis is the extension to three dimensions of a finite element two-phase formulation based on front-tracking with the interface uncoupled from the fluid mesh, combining the semi-Lagrangian method for treating the advective term of the Navier-Stokes equations, which is unconditionally stable and preserves matrix symmetry, with a finite element discretization of the Laplace-Beltrami operator to calculate the curvature of the interface. This combination allows straightforward numerical implementation, avoids the need for remeshing of the fluid mesh and offers accurate results for three-dimensional problems in diverse geometric configurations.

Chapter 3

Methodology

3.1 Navier-Stokes Equations

In this section, the equations that describe the motion of two immiscible fluids separated by an explicit interface are presented. For two-phase flow, two different formulations to describe motion can be adopted, in the first one, the "one-fluid" formulation, both fluid phases are treated as a single fluid with variable properties over the fluid domain, changing values at the interface. In the second option, each immiscible phase is considered a different fluid, and the equations governing their motion have proper boundary conditions at the interface. This work was developed using the "one-fluid" formulation, and the equations in this chapter are presented in this form. The equations exhibited in the following section are rewritten in non-dimensional form and expressed as functions of known dimensionless groups. Details regarding the development of the equations of fluid motion can be found in [47] and [48].

The behavior of isothermal fluid motion can be described by the conservation of mass and conservation of momentum equations. When the constitutive equation for a Newtonian fluid is replaced in the conservation of momentum equation, the Navier-Stokes equations are obtained. These equations can be further simplified for the case of an incompressible fluid, and this is the form given below, starting with the conservation of mass equation.

$$\nabla \cdot \mathbf{v} = 0 \tag{3.1}$$

where $\mathbf{v}$ represents velocity, followed by the conservation of momentum equation for a Newtonian, incompressible fluid. It is worth noting, that this derivation of the momentum equation does not assume constant properties, leading to fluid properties as functions of position, and the non-explicit Laplacian term.

$$\rho(\mathbf{x})\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = -\nabla p + \nabla \cdot \mu(\mathbf{x})(\nabla \mathbf{v} + \nabla \mathbf{v}^T) + \rho(\mathbf{x})\mathbf{g} + \mathbf{f} \tag{3.2}$$

In the equation above, t represents the time, p represents the pressure and $\rho(\mathbf{x})$ represents the fluid specific mass at the point $\mathbf{x}$. Likewise, $\mu(\mathbf{x})$ represents the fluid viscosity. $\mathbf{f}$ is the term representing body force, and $\mathbf{g}$ is the gravity force acting on the fluid. The term $\nabla \mathbf{v}^T$ is considered in the current model due to the variation of fluid properties across the interface.

For this work, the force due to surface tension force $\mathbf{f}$ will be inserted as a volumetric force, through the continuum surface force method, as done in [49]. The surface tension force can be represented by

$$\mathbf{f} = \sigma \kappa \delta \mathbf{n} \tag{3.3}$$

where σ is the surface tension coefficient, κ is the interface's mean curvature at a point, δ is a Dirac delta function with support at the interface, and $\mathbf{n}$ an outward normal to the interface. Replacing this term into the momentum equation, one finds

$$\rho(\mathbf{x})\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = -\nabla p + \nabla \cdot \mu(\mathbf{x})(\nabla \mathbf{v} + \nabla \mathbf{v}^T) + \rho(\mathbf{x})\mathbf{g} + \sigma \kappa \delta \mathbf{n} \tag{3.4}$$

The equations above can be rewritten into their non-dimensional form replacing their dimensional units with their non-dimensional form as given by

$$\begin{aligned}
&\mathbf{v}^* = \frac{\mathbf{v}}{V}, \quad \mathbf{x}^* = \frac{\mathbf{x}}{L}, \quad t^* = \frac{tV}{L}, \quad p^* = \frac{p}{\rho_d V^2}, \quad \rho^* = \frac{\rho}{\rho_d}, \quad \mu^* = \frac{\mu}{\mu_d}, \\
&\nabla^* = \nabla L, \quad \mathbf{g}^* = \frac{g}{g_o}, \quad \sigma^* = \frac{\sigma}{\sigma_o}, \quad \kappa^* = \kappa L, \quad \delta^* = \frac{\delta}{L}
\end{aligned} \tag{3.5}$$

where L and V refer to characteristic length and characteristic velocity, respectively. g_o and σ_o are the reference gravity acceleration and reference surface tension coefficient. The superscript $*$ refers to non-dimensional quantities, and the subscript d refers to the denser fluid property, be it specific mass or viscosity.

Replacing the non-dimensional parameters into the mass conservation equation, one obtains

$$\frac{V}{L}\nabla^* \cdot \mathbf{v}^* = 0 \tag{3.6}$$

which can then be multiplied by L/V to result in

$$\nabla^* \cdot \mathbf{v}^* = 0. \tag{3.7}$$

To obtain the non-dimensional momentum equation, one can proceed in the same fashion, applying the non-dimensional parameters to the momentum equation. Doing so, one obtains the following expression:

$$\begin{aligned}
&\rho^* \rho_d \left(\frac{\partial \mathbf{v}^*}{\partial t} \frac{V^2}{L} + \mathbf{v}^* \cdot \nabla^* \mathbf{v}^* \frac{V^2}{L} \right) = -\nabla^* p^* \frac{\rho_d V^2}{L} \\
&+ \frac{1}{L} \nabla^* \cdot \mu^* \mu_d \left(\nabla^* \mathbf{v}^* \frac{V}{L} + \nabla^* \mathbf{v}^{*T} \frac{V}{L} \right) + \rho(\mathbf{x})^* \mathbf{g}^* \rho_d g_o + \frac{1}{L^2} \sigma_o \sigma^* \kappa^* \delta^* \mathbf{n}.
\end{aligned} \tag{3.8}$$

The equation above can be multiplied by $L/\rho_d V^2$ to result in

$$\begin{aligned}\rho(\mathbf{x})^*\left(\frac{\partial \mathbf{v}^*}{\partial t}+\mathbf{v}^*\cdot\nabla\mathbf{v}^*\right) &= -\nabla^* p^* + \frac{\mu_d}{\rho_d V L}\nabla^*\cdot\mu(\mathbf{x})^*(\nabla^*\mathbf{v}^* + \nabla^*\mathbf{v}^{*T}) \\ &+ \frac{\rho(\mathbf{x})^*\mathbf{g}^* L}{V^2}\mathbf{g_o} + \frac{\sigma_o}{LV^2\rho_d}\sigma^*\kappa^*\delta^*\mathbf{n}.\end{aligned} \tag{3.9}$$

The terms $\sigma^*\kappa^*\delta^*\mathbf{n}$ can be grouped into $\mathbf{f}^*$. One can also recognize three dimensionless numbers expressed in the equation. The Reynolds number, the Weber number, and the Froude number.

The Reynolds number is given by the following expression:

$$Re = \frac{\rho_d V L}{\mu_d} \tag{3.10}$$

and can be understood as a ratio of advection and diffusion occurring on a flow, or a ratio of inertial effects and viscous effects.

The Weber number is given by the following expression:

$$We = \frac{\rho_d L V^2}{\sigma_o} \tag{3.11}$$

and it relates inertial effects to the surface tension effects. The surface tension coefficient, σ is taken as constant, excluding Marangoni effects acting on the interface.

The Froude number is given by the following expression:

$$Fr = \frac{V}{\sqrt{gL}} \tag{3.12}$$

and it relates the inertial effects to the effects provoked by an external field, in this case, gravity.

Replacing the surface tension force and the dimensionless numbers presented, the non-dimensional form of the momentum equation is obtained, given by the following equation:

$$\begin{aligned} &\rho(\mathbf{x})^* \left(\frac{\partial \mathbf{v}^*}{\partial t^*} + \mathbf{v}^* \cdot \nabla^* \mathbf{v}^* \right) = -\nabla^* p^* \\ &+ \frac{1}{Re} \nabla^* \cdot \mu^* [\nabla^* \mathbf{v}^* + (\nabla^* \mathbf{v}^*)^T] + \frac{1}{Fr^2} \rho(\mathbf{x})^* \mathbf{g}^* + \frac{1}{We} \mathbf{f}^* \end{aligned} \tag{3.13}$$

Gravity driven flows do not have a velocity value to take as reference. For those flows, gravity is used as reference, and different non-dimensional numbers are utilized, the Archimedes and Eötvös numbers, given respectively by:

$$Fr = 1 \qquad N = \frac{\rho_c^2 g_o D_o^3}{\mu_c} \qquad Eo = \frac{\rho_c g_o D_o^2}{\sigma_o} \tag{3.14}$$

where D_o is the bubble's diameter or the channel's diameter. The Archimedes number represents the weight of the gravitational forces against the viscous forces, and the Eötvös number is a ratio of gravitational forces to the surface tension force.

In a gravity driven flow, one can simply replace in Eq. 3.2 the values for $Re = N^{\frac{1}{2}}$, $We = Eo$ and $Fr = 1$. When studying Taylor bubbles, the liquid film thickness is a flow parameter of interest. It can be calculated using an equation derived from lubrication theory, which is a function of the Morton number, given by:

$$Mo = \frac{\mu_c^4 g_o}{\rho_c \sigma_o^3} = \frac{We^3}{FrRe^4} = \frac{Eo^3}{N^2} \tag{3.15}$$

3.2 Boundary and Initial Conditions

A well-posed problem modeled by a differential equation requires boundary conditions for a unique solution to be found from the set of possible solutions. The boundary condition is a constraint, a known value of the differential equation at a given point, that restricts the number of available solutions for that differential equation.

One type of boundary condition for a fluid problem is the flow condition when the fluid is in contact with an unmovable, solid surface. Disregarding a scenario where the solid is composed of a porous material, it is clear that the fluid cannot penetrate the solid. Therefore the fluid velocity perpendicular to the solid surface is zero. The velocity tangential to the solid surface is also null. This fact is confirmed experimentally, and it is attributed to the fluid's viscosity.

The initial conditions describe the state of the system, and the field variable values, at the starting point in time.

When solving a differential equation numerically, the boundary conditions and initial conditions should reflect the conditions described above. The boundary conditions can be classified in *Dirichlet* boundary conditions, where the solution value is prescribed at the boundary, or *Neumann* boundary condition which prescribes the value of the solution's derivative at the boundary, more details in [50].

The boundary conditions used in this work are described below:

- No-slip condition (or wall). The no-slip condition is a prescribed velocity condition, where all velocity component values are set to zero.

- Inflow condition. The inflow condition is a prescribed velocity condition, setting the velocity values to the desired values. This is intended to represent a mass influx into the problem's domain.

- Outflow condition. The outflow condition is a prescribed pressure condition, where the pressure values are set to zero. This represents a mass flow outside the problem's domain.

- Symmetry condition. The symmetry condition imposes a null velocity value in the direction normal to the symmetry plane. The tangential velocity components are allowed to be non-null. The symmetry condition is useful in situations where, if the geometry permits, it is convenient to represent only part of the problem's domain, saving on computational costs. Symmetry boundary conditions are applicable not only to symmetric problems but also to problems where the geometry is periodic.

3.3 Finite Element Method

The finite element method is a computational technique used to solve differential and integral equations. it is widely used in engineering to solve problems in fluid dynamics, heat transfer, structural analysis, and more, as it can provide a very accurate approximate solution where a direct analytical solution would be difficult to obtain.

The method consists of dividing the problem domain into discrete finite-sized partitions called elements which are interconnected at specific points called nodes. These elements can be of a variety of shapes, in one, two, or three dimensions, for example, line segments, triangles, quadrilaterals, tetrahedrons, and others [51]. The finite element method uses a weighted residual formulation or variational formulation of the differential equation to obtain an approximation of its solution at the nodes. This formulation can be obtained by multiplying the equation by a test function and integrating it over the entire domain of the problem. The solution error does not need to be zero throughout the entire domain. Instead, the integral of the error, also known as the residuum, must vanish to achieve a solution.

The finite element method can solve problems with complex geometries and boundary conditions because it does not require a specific shape for the finite elements, allowing for the representation of irregularly shaped domains and varying boundary conditions.

The starting point is the Navier-Stokes equations for incompressible flow, as detailed in the previous section, where the * superscript was dropped for convenience

$$\nabla \cdot \mathbf{v} = 0 \tag{3.16}$$

$$\rho(\mathbf{x})\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = -\nabla p + \frac{1}{Re}\nabla \cdot \mu(\mathbf{x})[\nabla \mathbf{v} + (\nabla \mathbf{v})^T] + \frac{1}{Fr^2}\rho(\mathbf{x})\mathbf{g} + \frac{1}{We}\mathbf{f} \tag{3.17}$$

which are valid for an arbitrary domain $\Omega \subset R^m$, presented in fig. 3.1 , and the following initial and boundary conditions

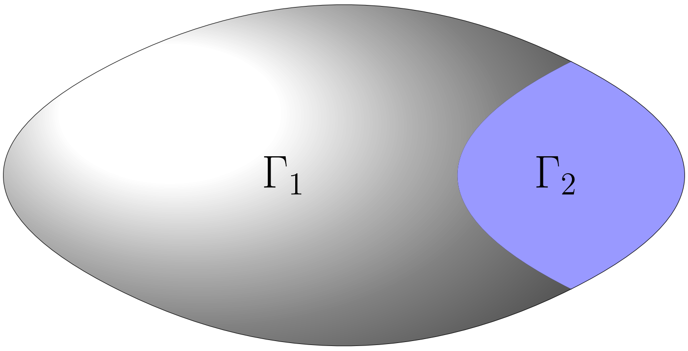


Figure 3.1: Three dimensional arbitrary domain Ω, with boundaries Γ_1 and Γ_2 visible on its surface, represented in gray and blue, respectively

$$\mathbf{v} = \mathbf{v}_\Gamma \quad and \quad \nabla p \cdot \mathbf{n} = 0 \quad in \quad \Gamma_1 \tag{3.18a}$$

$$(\mathbf{n} \cdot \nabla)\mathbf{v} \cdot \mathbf{n} = 0 \quad and \quad p = 0 \quad in \quad \Gamma_2 \tag{3.18b}$$

$$\mathbf{v}_{t=0} = p_{t=0} = 0 \tag{3.18c}$$

where $\mathbf{v}_{t=0}$ represents the initial velocity at the points that do not belong to Γ_1 or Γ_2 and $\mathbf{v}_{\boldsymbol{\Gamma}}$ the prescribed velocity values at the boundary domain. These equations can be used to represent the previously described boundary conditions. For example:

- The inflow condition can be obtained by setting $v_x = 1, v_y = 0, v_z = 0 \quad and \quad \nabla p \cdot \mathbf{n} = 0$.
- The no-slip condition can be obtained by setting $\mathbf{v} = 0 \quad and \quad \nabla p \cdot \mathbf{n} = 0$.
- The outflow condition can be obtained by setting $(\mathbf{n} \cdot \nabla)\mathbf{v} \cdot \mathbf{n} = 0 \quad and \quad p = 0$.

Multiplying the equations 3.16 and 3.17 by weighting functions $\mathbf{w}$ and $\mathbf{q}$, associated to velocity and pressure respectively, results in

$$\begin{aligned} &\int_\Omega \Big\{ \rho(\mathbf{x}) \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) + \nabla p - \frac{1}{Re} \nabla \cdot \mu(\mathbf{x}) [\nabla \mathbf{v} + (\nabla \mathbf{v})^T] \\ &- \frac{1}{Fr^2} \rho(\mathbf{x}) \mathbf{g} - \frac{1}{We} \mathbf{f} \Big\} \cdot \mathbf{w} d\Omega = 0 \end{aligned} \tag{3.19}$$

$$\int_\Omega (\nabla \cdot \mathbf{v}) \cdot \mathbf{q} d\Omega = 0 \tag{3.20}$$

which can be added up and written as

$$\begin{aligned}\int_\Omega \rho(\mathbf{x})\left\{\frac{\partial \mathbf{v}}{\partial t}+\mathbf{v}\cdot\nabla\mathbf{v}\right\}\cdot\mathbf{w}d\Omega \quad &+\int_\Omega \nabla p\cdot\mathbf{w}d\Omega \\ -\int_\Omega\left\{\frac{1}{Re}\nabla\cdot\mu(\mathbf{x})[\nabla\mathbf{v}+\nabla\mathbf{v}^T]\right\}\cdot\mathbf{w}d\Omega &- \int_\Omega \frac{1}{Fr^2}\rho(\mathbf{x})\mathbf{g}\cdot\mathbf{w}d\Omega \\ -\int_\Omega \frac{1}{We}\mathbf{f}\cdot\mathbf{w}d\Omega+\int_\Omega(\nabla\cdot\mathbf{v})\cdot\mathbf{q}d\Omega = 0\end{aligned} \tag{3.21}$$

The first term on equation 3.21 is non-linear and will be treated separately by a first-order semi-Lagrangian method. It can be written as

$$\int_\Omega \rho(\mathbf{x})\left\{\frac{\partial \mathbf{v}}{\partial t}+\mathbf{v}\cdot\nabla\mathbf{v}\right\}\cdot\mathbf{w}d\Omega = \int_\Omega \rho(\mathbf{x})\frac{D\mathbf{v}}{Dt}\cdot\mathbf{w}d\Omega \tag{3.22}$$

The third term on equation 3.21 contains a second-order derivative, thus it requires that the interpolation functions be quadratic or higher order so that they can be differentiated twice. To address this issue, Green's Theorem can be employed to reduce the second-order derivative to a lower-order term, thus reducing the demand for high-order functions. This is a standard technique used in the finite element method and can be found in classical literature such as [51]. Firstly, the term can be separated as follows

$$\int_\Omega\left\{\nabla\cdot\mu(\mathbf{x})[\nabla\mathbf{v}+\nabla\mathbf{v}^T]\right\}\cdot\mathbf{w}d\Omega = \int_\Omega[\nabla\cdot\mu(\mathbf{x})\nabla\mathbf{v}]\cdot\mathbf{w}d\Omega+\int_\Omega[\nabla\cdot\mu(\mathbf{x})\nabla\mathbf{v}^T]\cdot\mathbf{w}d\Omega \tag{3.23}$$

Green's theorem for vector fields, where $\Omega \subset R^m$, $\Gamma = \partial\Omega$ is the boundary of domain Ω, and $\mathbf{u},\mathbf{w} : U \subset R^m \rightarrow R^m$ are vectors fields, is given by

$$\int_\Omega \nabla^2 \mathbf{u} \cdot \mathbf{w} d\Omega = \int_\Gamma \mathbf{n} \cdot (\nabla \mathbf{u} \cdot \mathbf{w}) d\Gamma - \int_\Omega (\nabla \mathbf{u} : \nabla \mathbf{w}^T) d\Omega. \tag{3.24}$$

Applying it on both terms on the right side of equation 3.23, results in

$$\int_\Omega [\nabla \cdot \mu(\mathbf{x}) \nabla \mathbf{v}] \cdot \mathbf{w} d\Omega = \int_\Gamma \mu(\mathbf{x}) \mathbf{n} \cdot (\nabla \mathbf{v} \cdot \mathbf{w}) d\Gamma - \int_\Omega \mu(\mathbf{x}) \nabla \mathbf{v} : \nabla \mathbf{w}^T d\Omega \tag{3.25}$$

and

$$\int_\Omega [\nabla \cdot \mu(\mathbf{x}) \nabla \mathbf{v}^T] \cdot \mathbf{w} d\Omega = \int_\Gamma \mu(\mathbf{x}) \mathbf{n} \cdot (\nabla \mathbf{v}^T \cdot \mathbf{w}) d\Gamma - \int_\Omega \mu(\mathbf{x}) \nabla \mathbf{v}^T : \nabla \mathbf{w}^T d\Omega \tag{3.26}$$

which can be summed and replaced back into equation 3.23 as

$$\begin{aligned} \int_\Omega \nabla \cdot \mu(\mathbf{x}) [\nabla \mathbf{v} + \nabla \mathbf{v}^T] \cdot \mathbf{w} d\Omega = \int_\Gamma \mathbf{n} \cdot [\mu(\mathbf{x}) (\nabla \mathbf{v} + \nabla \mathbf{v}^T) \cdot \mathbf{w}] d\Gamma \\ - \int_\Omega \mu(\mathbf{x}) [(\nabla \mathbf{v} + \nabla \mathbf{w}^T) : \nabla \mathbf{w}^T] d\Omega \end{aligned} \tag{3.27}$$

As presented in [52], the weight function $\mathbf{w}$ is required to be zero at the domain's Dirichlet's boundary conditions, and so the first term on the right side of equation 3.27 is null. The same procedure can be applied to the pressure term

$$\int_\Omega \nabla p \cdot \mathbf{w} d\Omega = - \int_\Omega p \nabla \cdot \mathbf{w} d\Omega + \int_\Gamma p \mathbf{w} \cdot \mathbf{n} d\Omega \tag{3.28}$$

where the boundary term also becomes null. Replacing both results above in 3.21, gives the following equation

$$\begin{aligned}\int_\Omega \rho(\mathbf{x})\frac{D\mathbf{v}}{Dt}\cdot\mathbf{w}d\Omega \quad &- \int_\Omega p\cdot\nabla\mathbf{w}d\Omega \quad + \frac{1}{Re}\int_\Omega \mu(\mathbf{x})(\nabla\mathbf{v}+\nabla\mathbf{v}^T):\nabla\mathbf{w}d\Omega \\ -\frac{1}{Fr^2}\int_\Omega \rho(\mathbf{x})\mathbf{g}\cdot\mathbf{w}d\Omega \quad &- \frac{1}{We}\int_\Omega \mathbf{f}\cdot\mathbf{w}d\Omega \quad + \int_\Omega (\nabla\cdot\mathbf{v})\cdot\mathbf{q}d\Omega = 0\end{aligned} \tag{3.29}$$

3.4 Element Discretization

It was stated at the start of the chapter that the finite element method divides the spatial domain into several finite-sized elements, this division turns the continuum strong form of the problem into a discrete system of equations.

The discrete solution for the velocity components, pressure, and the gravity and surface tension force can be expressed in terms of shape functions and their values at the finite element nodes by

$$\begin{aligned} u(\mathbf{x}) &\approx u^e(\mathbf{x}) = \sum_{i=1}^{n} u_i^e N_i(\mathbf{x}) \\ v(\mathbf{x}) &\approx v^e(\mathbf{x}) = \sum_{i=1}^{n} v_i^e N_i(\mathbf{x}) \\ w(\mathbf{x}) &\approx w^e(\mathbf{x}) = \sum_{i=1}^{n} w_i^e N_i(\mathbf{x}) \\ p(\mathbf{x}) &\approx p^e(\mathbf{x}) = \sum_{j=1}^{m} p_j^e L_j(\mathbf{x}) \\ g(\mathbf{x}) &\approx g^e(\mathbf{x}) = \sum_{i=1}^{m} g_i^e N_i(\mathbf{x}) \end{aligned} \tag{3.30}$$

where $N_i(\mathbf{x})$ and $L_j(\mathbf{x})$ are approximation functions used to interpolate the values of the velocity components u, v, w, the pressure p and the gravitational force g inside a finite element. These are called interpolation functions or shape functions and they are commonly polynomials because they can be easily derived from interpolation theory and give exact results when integrated numerically. They must also respect the following condition

$$N_i^e(\mathbf{x_j}) = L_i^e(\mathbf{x_j}) = \delta_i j \tag{3.31}$$

which means that at a given node i, the value of N_i must be one, and zero at all other nodes, the same rule applies for L_i.

3.5 Finite Element Discretization of the Navier Stokes Equations

To complete the finite element discretization of the Navier Stokes equations, one replaces the values of $\mathbf{u}$, p, g and f in the weak form displayed in 3.29 for the approximations expressed in 3.30. In this step, the surface tension term will be omitted, its discretization will be described separately in the following section. The discretization process is much easier detailed when the quantities are expressed in index notation, so the weak form is expressed as

$$\begin{aligned}
&\int_\Omega \rho \left[\frac{Du}{Dt} w_x + \frac{Dv}{Dt} w_y + \frac{Dw}{Dt} w_z \right] d\Omega \quad - \int_\Omega p \left[\frac{\partial w_x}{\partial x} + \frac{\partial w_y}{\partial y} + \frac{\partial w_z}{\partial z} \right] d\Omega \\
&\quad + \frac{1}{Re} \int_\Omega \mu \Bigg[\bigg(\frac{\partial u}{\partial x}\frac{\partial w_x}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial w_y}{\partial x} + \frac{\partial u}{\partial z}\frac{\partial w_z}{\partial x} + \frac{\partial v}{\partial x}\frac{\partial w_x}{\partial y} \\
&+ \frac{\partial v}{\partial y}\frac{\partial w_y}{\partial y} + \frac{\partial v}{\partial z}\frac{\partial w_z}{\partial y} + \frac{\partial w}{\partial x}\frac{\partial w_x}{\partial z} + \frac{\partial w}{\partial y}\frac{\partial w_y}{\partial z} + \frac{\partial w}{\partial z}\frac{\partial w_z}{\partial z} \bigg) \\
&+ \bigg(\frac{\partial u}{\partial x}\frac{\partial w_x}{\partial x} + \frac{\partial v}{\partial x}\frac{\partial w_y}{\partial x} + \frac{\partial w}{\partial x}\frac{\partial w_z}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial w_x}{\partial y} \\
&+ \frac{\partial v}{\partial y}\frac{\partial w_y}{\partial y} + \frac{\partial w}{\partial y}\frac{\partial w_z}{\partial y} + \frac{\partial u}{\partial z}\frac{\partial w_x}{\partial z} + \frac{\partial v}{\partial z}\frac{\partial w_y}{\partial z} + \frac{\partial w}{\partial z}\frac{\partial w_z}{\partial z} \bigg) \Bigg] d\Omega \\
&\quad - \frac{1}{Fr^2} \int_\Omega \rho (g_x w_x + g_y w_y + g_z w_z) d\Omega = 0
\end{aligned} \tag{3.32}$$

where u, v and w are the velocity components on the x, y and z directions, similarly,

w_x, w_y and w_z are the weight function components in the Cartesian directions. Note that one can express the above equation as one equation for each of the velocity components, including the mass conservation equation, and if all three equations are satisfied, then the above equation is automatically satisfied

$$\begin{aligned}&\int_\Omega \rho \frac{Du}{Dt} w_x d\Omega \quad - \int_\Omega p \frac{\partial w_x}{\partial x} d\Omega \quad + \frac{1}{Re} \int_\Omega \mu \bigg[\left(\frac{\partial u}{\partial x}\frac{\partial w_x}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial w_y}{\partial x} + \frac{\partial u}{\partial z}\frac{\partial w_z}{\partial x} \right) \\ &+ \left(\frac{\partial u}{\partial x}\frac{\partial w_x}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial w_x}{\partial y} + \frac{\partial u}{\partial z}\frac{\partial w_x}{\partial z} \right) \bigg] d\Omega \\ &\quad - \frac{1}{Fr^2} \int_\Omega \rho g_x w_x d\Omega + \int_\Omega \frac{\partial u}{\partial x} q d\Omega. = 0\end{aligned} \tag{3.33}$$

$$\begin{aligned}&\int_\Omega \rho \frac{Dv}{Dt} w_y d\Omega \quad - \int_\Omega p \frac{\partial w_y}{\partial y} d\Omega \quad + \frac{1}{Re} \int_\Omega \mu \bigg[\left(\frac{\partial v}{\partial x}\frac{\partial w_x}{\partial y} + \frac{\partial v}{\partial y}\frac{\partial w_y}{\partial y} + \frac{\partial v}{\partial z}\frac{\partial w_z}{\partial y} \right) \\ &+ \left(\frac{\partial v}{\partial x}\frac{\partial w_y}{\partial x} + \frac{\partial v}{\partial y}\frac{\partial w_y}{\partial y} + \frac{\partial v}{\partial z}\frac{\partial w_y}{\partial z} \right) \bigg] d\Omega \\ &\quad - \frac{1}{Fr^2} \int_\Omega \rho g_y w_y d\Omega + \int_\Omega \frac{\partial v}{\partial y} q d\Omega. = 0\end{aligned} \tag{3.34}$$

$$\begin{aligned}&\int_\Omega \rho \frac{Dw}{Dt} w_z d\Omega \quad - \int_\Omega p \frac{\partial w_z}{\partial z} d\Omega \quad + \frac{1}{Re} \int_\Omega \mu \bigg[\left(\frac{\partial w}{\partial x}\frac{\partial w_x}{\partial z} + \frac{\partial w}{\partial y}\frac{\partial w_y}{\partial z} + \frac{\partial w}{\partial z}\frac{\partial w_z}{\partial z} \right) \\ &+ \left(\frac{\partial w}{\partial x}\frac{\partial w_z}{\partial x} + \frac{\partial w}{\partial y}\frac{\partial w_z}{\partial y} + \frac{\partial w}{\partial z}\frac{\partial w_z}{\partial z} \right) \bigg] d\Omega \\ &\quad - \frac{1}{Fr^2} \int_\Omega \rho g_z w_z d\Omega + \int_\Omega \frac{\partial w}{\partial z} q d\Omega. = 0\end{aligned} \tag{3.35}$$

The equations described above must hold true for any choice of the weight functions w and q. One choice for the weight functions is to use the same functions used to interpolate the velocity components and the pressure inside the element:

$$w_x = w_y = w_z = \sum_{k=1}^{n} N_k(\mathbf{x}) \qquad\qquad q = \sum_{j=1}^{m} L_j(\mathbf{x}). \tag{3.36}$$

This is called the Galerkin method. Replacing the approximation for the weight functions **w** given in 3.36 into equations 3.33 through 3.35, along with the approximations in 3.30.

For brevity, the summation sign will be omitted in the following equations. The result of the replacement, given that $DN_i/Dt = 0$, is given bellow

$$\begin{aligned}
&\int_\Omega \rho \frac{Du_i}{Dt} N_i N_k d\Omega \quad - \int_\Omega L_j p_j \frac{\partial N_k}{\partial x} d\Omega \\
&\quad + \frac{1}{Re} \int_\Omega \mu \Bigg[\left(\frac{\partial N_i}{\partial x} u_i \frac{\partial N_k}{\partial x} + \frac{\partial N_i}{\partial y} u_i \frac{\partial N_k}{\partial x} + \frac{\partial N_i}{\partial z} u_i \frac{\partial N_k}{\partial x} \right) \\
&+ \left(\frac{\partial N_i}{\partial x} u_i \frac{\partial N_k}{\partial x} + \frac{\partial N_i}{\partial y} u_i \frac{\partial N_k}{\partial y} + \frac{\partial N_i}{\partial z} u_i \frac{\partial N_k}{\partial z} \right) \Bigg] d\Omega \\
&\quad - \frac{1}{Fr^2} \int_\Omega \rho g_x N_k d\Omega + \int_\Omega \frac{\partial N_k}{\partial x} u_j L_j d\Omega = 0
\end{aligned} \tag{3.37}$$

$$\begin{aligned}
&\int_\Omega \rho \frac{Dv_i}{Dt} N_i N_k d\Omega \quad - \int_\Omega L_j p_j \frac{\partial N_k}{\partial y} d\Omega \\
&\quad + \frac{1}{Re} \int_\Omega \mu \Bigg[\left(\frac{\partial N_i}{\partial x} v_i \frac{\partial N_k}{\partial y} + \frac{\partial N_i}{\partial y} v_i \frac{\partial N_k}{\partial y} + \frac{\partial N_i}{\partial z} v_i \frac{\partial N_k}{\partial y} \right) \\
&+ \left(\frac{\partial N_i}{\partial x} v_i \frac{\partial N_k}{\partial x} + \frac{\partial N_i}{\partial y} v_i \frac{\partial N_k}{\partial y} + \frac{\partial N_i}{\partial z} v_i \frac{\partial N_k}{\partial z} \right) \Bigg] d\Omega \\
&\quad - \frac{1}{Fr^2} \int_\Omega \rho g_y N_k d\Omega + \int_\Omega \frac{\partial N_k}{\partial y} v_j L_j d\Omega = 0
\end{aligned} \tag{3.38}$$

$$\begin{aligned}
&\int_\Omega \rho \frac{Dw_i}{Dt} N_i N_k d\Omega \quad - \int_\Omega L_j p_j \frac{\partial N_k}{\partial z} d\Omega \\
&\quad + \frac{1}{Re} \int_\Omega \mu \Bigg[\left(\frac{\partial N_i}{\partial x} w_i \frac{\partial N_k}{\partial z} + \frac{\partial N_i}{\partial y} w_i \frac{\partial N_k}{\partial z} + \frac{\partial N_i}{\partial z} w_i \frac{\partial N_k}{\partial z} \right) \\
&+ \left(\frac{\partial N_i}{\partial x} w_i \frac{\partial N_k}{\partial x} + \frac{\partial N_i}{\partial y} w_i \frac{\partial N_k}{\partial y} + \frac{\partial N_i}{\partial z} w_i \frac{\partial N_k}{\partial z} \right) \Bigg] d\Omega \\
&\quad - \frac{1}{Fr^2} \int_\Omega \rho g_z N_k d\Omega + \int_\Omega \frac{\partial N_k}{\partial z} w_j L_j d\Omega = 0
\end{aligned} \tag{3.39}$$

The continuity equation is linked to the pressure and thus the terms derived from it are evaluated at the same nodes. The weight function for the continuity equation uses the same interpolation function as the pressure. The terms of the equations can be represented by

$$\begin{aligned}
m &= \sum_{i=1}^{n}\sum_{k=1}^{n}\int_{\Omega^e} N_i N_k d\Omega^e & q_x &= \sum_{j=1}^{m}\sum_{k=1}^{n}\int_{\Omega^e} L_j \frac{\partial N_k}{\partial x} d\Omega^e \\
q_y &= \sum_{j=1}^{m}\sum_{k=1}^{n}\int_{\Omega^e} L_j \frac{\partial N_k}{\partial y} d\Omega^e & q_z &= \sum_{j=1}^{m}\sum_{k=1}^{n}\int_{\Omega^e} L_j \frac{\partial N_k}{\partial z} d\Omega^e \\
d_x &= \sum_{k=1}^{n}\sum_{j=1}^{m}\int_{\Omega^e} L_k \frac{\partial N_j}{\partial x} d\Omega^e & d_y &= \sum_{k=1}^{n}\sum_{j=1}^{m}\int_{\Omega^e} L_k \frac{\partial N_j}{\partial y} d\Omega^e \\
d_z &= \sum_{k=1}^{n}\sum_{j=1}^{m}\int_{\Omega^e} L_k \frac{\partial N_j}{\partial z} d\Omega^e & k_{xx} &= \sum_{i=1}^{n}\sum_{k=1}^{n}\int_{\Omega^e} \frac{\partial N_i}{\partial x}\frac{\partial N_k}{\partial x} d\Omega^e \\
k_{yy} &= \sum_{i=1}^{n}\sum_{k=1}^{n}\int_{\Omega^e} \frac{\partial N_i}{\partial y}\frac{\partial N_k}{\partial y} d\Omega^e & k_{zz} &= \sum_{i=1}^{n}\sum_{k=1}^{n}\int_{\Omega^e} \frac{\partial N_i}{\partial z}\frac{\partial N_k}{\partial z} d\Omega^e \\
k_{xy} &= \sum_{i=1}^{n}\sum_{k=1}^{n}\int_{\Omega^e} \frac{\partial N_i}{\partial x}\frac{\partial N_k}{\partial y} d\Omega^e & k_{yx} &= \sum_{i=1}^{n}\sum_{k=1}^{n}\int_{\Omega^e} \frac{\partial N_i}{\partial y}\frac{\partial N_k}{\partial x} d\Omega^e \\
k_{xz} &= \sum_{i=1}^{n}\sum_{k=1}^{n}\int_{\Omega^e} \frac{\partial N_i}{\partial x}\frac{\partial N_k}{\partial z} d\Omega^e & k_{zx} &= \sum_{i=1}^{n}\sum_{k=1}^{n}\int_{\Omega^e} \frac{\partial N_i}{\partial z}\frac{\partial N_k}{\partial x} d\Omega^e \\
k_{yz} &= \sum_{i=1}^{n}\sum_{k=1}^{n}\int_{\Omega^e} \frac{\partial N_i}{\partial y}\frac{\partial N_k}{\partial z} d\Omega^e & k_{zy} &= \sum_{i=1}^{n}\sum_{k=1}^{n}\int_{\Omega^e} \frac{\partial N_i}{\partial z}\frac{\partial N_k}{\partial y} d\Omega^e
\end{aligned} \tag{3.40}$$

Each of the terms above results in a matrix of dimensions ik, except for the g matrices, which have kj dimension, and d, which have jk. It is important to note that

$$k_{xy} = k_{yz}^T, \qquad k_{xz} = k_{zx}^T \qquad and \qquad k_{yz} = k_{zy}^T \tag{3.41}$$

3.6 Discrete Surface Tension Force

The last term on equation 3.29 can be expanded to

$$\frac{1}{We}\int_{\Omega} \sigma\delta\kappa\mathbf{n}\cdot\mathbf{w}d\Omega \tag{3.42}$$

where σ is the surface tension coefficient, κ is the mean curvature, δ is the Dirac delta function and $\mathbf{n}$ is the normal vector. The Dirac delta function is used as a device to indicate that the surface tension exists only on the interface. It can be embedded in the integral, by changing the integral limit, resulting in

$$-\frac{1}{We}\int_{\Gamma}\sigma\kappa\mathbf{n}\cdot\mathbf{w}d\Gamma \tag{3.43}$$

It is described in [53] that there is an operator that maps a point $\mathbf{x}$ of a given surface to a vector $\mathbf{C}(\mathbf{x})$, obeying the following relation

$$\mathbf{C}(\mathbf{x}) = \kappa\mathbf{n} = \nabla_s^2(\mathbf{x}) = \nabla_s\cdot(\nabla_s\mathbf{x}) \tag{3.44}$$

where ∇_s is the gradient operator over surfaces and ∇_s^2 is the Laplacian operator over surfaces. This operator is known as the Laplace-Beltrami operator for a given surface. The application of the Laplace-Beltrami operator on any point of the surface will produce its mean curvature at that point.

Moreover, as stated in [53], the Laplace-Beltrami operator is a generalization of the Laplacian operator from flat spaces to manifolds. For Euclidean space, the Laplace-Beltrami operator turns into the Laplacian operator. It is then possible to represent equation 3.43 by

$$-\frac{1}{We}\int_{\Gamma}\sigma\nabla_s^2\mathbf{x}\cdot\mathbf{w}d\Gamma \tag{3.45}$$

The surface tension coefficient was made non-dimensional by the Weber number, and consequently can be treated as a constant, $\sigma = 1$. Applying Green's identity to equation 3.45

$$-\int_{\Gamma} \nabla_s^2 \mathbf{x} \cdot \mathbf{w} d\Gamma = \int_{\Gamma} (\nabla_s \mathbf{x} : \nabla_s w) d\Gamma - \int_{C} (\nabla_s \mathbf{x} \cdot \mathbf{w}) \cdot \mathbf{n} dC \tag{3.46}$$

The second term on the right is an integral on the surface's contour. The surfaces presented in this work, due to their nature as bubbles or droplets, are all closed surfaces and therefore have no contour, therefore the second term on the right-hand side is null, resulting in

$$\int_{\Gamma} (\nabla_s \mathbf{x} : \nabla_s \mathbf{w}) d\Gamma \quad = \int_{\Gamma} \left(\frac{dx}{dx}\frac{dw_x}{dx} + \frac{dx}{dy}\frac{dw_x}{dy} + \frac{dy}{dx}\frac{dw_y}{dx} + \frac{dy}{dy}\frac{dw_y}{dy} \right) d\Gamma \tag{3.47}$$

At this point, the interface can be discretized into triangular elements, as presented in fig 3.4, of a of a three dimensional surface. The Galerkin finite element method can be applied to the equation 3.47, using the following approximated values over each element

$$x = \sum_{n=1}^{i} x_i^e N_i(\mathbf{x}) \qquad\qquad y = \sum_{n=1}^{i} y_i^e N_i(\mathbf{x}) \tag{3.48}$$

and

$$w_x = w_y = \sum_{n=1}^{j} N_j(\mathbf{x}) \tag{3.49}$$

where the shape functions $N(\mathbf{x})$ are the same for the coordinates and for the weight functions. Again, the shape functions utilized are linear triangle shape functions, presented in section 3.9. Taking into account that $\mathbf{x}_i^e$ is constant, replacing equations 3.48 and 3.49 results in

$$\begin{aligned}\int_\Gamma \nabla_s \mathbf{x} : \nabla_s \mathbf{w} d\Gamma = \int_\Gamma \bigg(& x_i \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + x_i \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \\ & + y_i \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + y_i \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \bigg) d\Gamma\end{aligned} \tag{3.50}$$

For brevity, the terms in the equation above can be represented as

$$\begin{aligned} c_{\xi\xi} &= \int_\Gamma \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} d\Gamma \\ c_{\eta\eta} &= \int_\Gamma \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} d\Gamma \end{aligned} \tag{3.51}$$

and the operator C, the finite element discretization of the Laplacian operator for a surface, defined as

$$C = c_{\xi\xi} + c_{\eta\eta} \tag{3.52}$$

The terms on the right, $c_{\xi\xi}$ and $c_{\eta\eta}$ represent the matrices that form the Laplacian operator for the linear triangular element.

The operator C presented above is defined for a local coordinates system, and cannot be directly applied to the surface inserted in a three-dimensional space. A transformation of local two-dimensional coordinates into global three-dimensional coordinates is necessary when assembling the operator. The local two-dimensional coordinate system can be observed in fig. 3.2, where ξ and η are defined locally in two dimensions, and the points i, j and k are defined in three-dimensional space.

The ξ unit vector can be defined as the normalized difference between two points, the vertices i and j in fig. 3.2. This vector is defined by

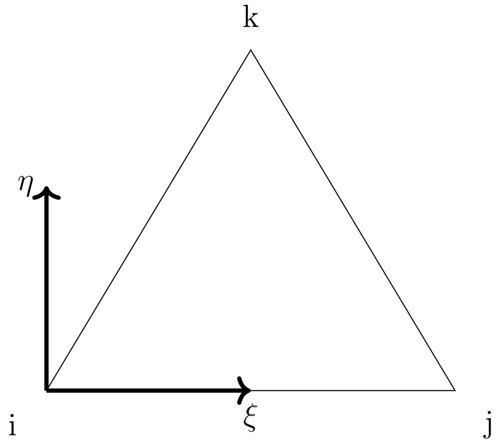


Figure 3.2: Local two-dimensional coordinate system, plotted over a triangle ijk. The points i, j and k are defined in three dimensions

$$\xi = \frac{\mathbf{x_j} - \mathbf{x_i}}{||\mathbf{x_j} - \mathbf{x_i}||} \tag{3.53}$$

The η unit vector is required to be perpendicular to both the triangle surface and the unit vector ξ. To achieve this, the definition given below can be used

$$\eta = \frac{[(\mathbf{x_j} - \mathbf{x_i}) \times (\mathbf{x_k} - \mathbf{x_i})] \times (\mathbf{x_j} - \mathbf{x_i})}{||(\mathbf{x_j} - \mathbf{x_i}) \times (\mathbf{x_k} - \mathbf{x_i})] \times (\mathbf{x_j} - \mathbf{x_i})||} \tag{3.54}$$

where the first cross product in equation 3.54 generates a vector perpendicular to the triangle's surface. The second cross product, between the resulting vector of the first cross product and the $(\mathbf{x_j} - \mathbf{x_i})$ subtraction results in a vector that is perpendicular to both the triangle's surface and the unit vector ξ. The division by the modulus turns it into a unit vector.

Once the coordinate transformation is done, the Laplacian operator can be applied to the Cartesian coordinates of the mesh element nodes, obtaining the vector coordinates.

$$
\begin{aligned}
C_x &= Cx \\
C_y &= Cy \\
C_z &= Cz
\end{aligned}
\tag{3.55}
$$

where x, y and z are the nodal Cartesian coordinates.

A finite element triangle mesh is composed of flat surfaces, where the surface normal changes abruptly at the edges. If one were to measure the curvature along a finite element mesh, one would find regions of zero curvature, followed by "spikes" due to the abrupt change. Clearly, this is not desired, since the interfaces the triangle mesh is representing are smooth surfaces. To complete the discretization and obtain the proper curvature values, it is necessary to divide the curvature values by the area around each node. To achieve this, the following is done:

$$\mathbf{C}(\mathbf{x})_{\mathbf{t}} = \sqrt{C_x^2 + C_y^2 + C_z^2} \tag{3.56}$$

and then dividing the $C(\mathbf{x})_t$ by the nodal area, the calculation of which is presented in a subsequent section, to obtain the nodal curvature:

$$\kappa = \frac{C(\mathbf{x})_t}{A_i} \tag{3.57}$$

where A_i is the area around node i. The surface tension term is then represented by

$$\mathbf{f} = -\frac{1}{We}[\kappa \mathbf{n}] \tag{3.58}$$

this representation was derived by integrating over the fluid interface. it is not applicable to the whole fluid domain. This was achieved by embedding the Dirac delta function into the integral, however, this is not desirable when adding the surface force term to the Navier-Stokes equations, and thus it is necessary to make the Dirac delta explicit again. One way to accomplish this is to use the Heaviside function gradient. The gradient of the Heaviside function, described in the next section, serves as a Dirac delta based on the interface and points in the normal direction, therefore

$$\mathbf{f} = -\frac{\kappa}{We}[\nabla H_x\mathbf{i} + \nabla H_y\mathbf{j} + \nabla H_z\mathbf{k}] \tag{3.59}$$

and this equation can be replaced directly into the Navier-Stokes equations discretization.

3.6.1 Surface Tension Force Assignment

The surface tension force is distributed from the interface mesh to the fluid mesh through the CSF method [49]. In this method, the surface tension force which exists only on the interface is transformed into a volumetric force acting upon the whole domain, but with non-null values only on nodes located at the interface by means of a Dirac delta function. This transformation is made simply by selecting a fluid mesh node, searching for the closest interface mesh node, and assigning the curvature value from the interface node to the fluid mesh node.

While this method is sufficiently accurate, in simulations where spurious oscillations appear, the interface mesh might suffer some warping or distortion, and an interface node can have its curvature value modified in a way that does not represent the region's curvature, reflecting upon surface tension value calculated from it, and affecting the simulation stability and accuracy. To alleviate these matters, in this work, the surface tension force of a given fluid mesh node is calculated using the curvature of the closest interface node like in the CSF method, but with an added contribution from its neighbor nodes, following the formula given by

$$\kappa_{fluid} = a\kappa_{interface} + \frac{b}{n}\sum_{i=n}\kappa_{neighbor} \tag{3.60}$$

where κ is the curvature, n is the number of neighbours an interface node has, which are the nodes present in elements the node belongs to, or the nodes resulting from the umbrella operator applied to the analyzed node, and a and b are values between 0 and 1, typically, $a = b = 0.5$.

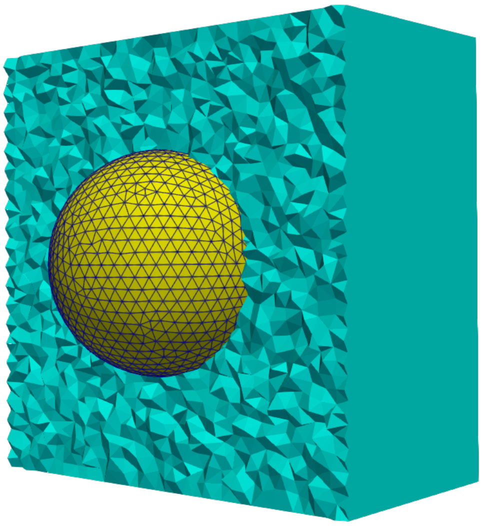


Figure 3.3: Representation of a transversal cut of a fluid mesh preserving tetrahedra faces with an interface mesh positioned at its center.

3.6.2 Fluid Interface Representation

When simulating two-phase flow, it is crucial to have some way of representing the interface separating the immiscible fluids. Besides having the role of identifying the two fluid phases, this interface is used to calculate the surface tension effects through its curvature. In this work the fluid interface is represented by a closed surface inserted in three dimensional space, constructed by triangular surface elements, as presented in fig. 3.3. The nodes belonging to this interface mesh are not shared by the fluid mesh. The coupling between the interface mesh and the fluid mesh is effectuated when the interface mesh's nodes are advected by the fluid mesh velocity fields, and the surface tension force, inputted as a source term in the Navier-Stokes equation, is calculated through the interface mesh's curvature, obtained using the Laplace-Beltrami operator.

The identification of the fluid phases is done through the use of the Heaviside function, anchored at the interface mesh. In the finite element matrix assembly process, if the interface "cuts through" an element, the properties assigned to that element are an average from its nodal values.

The abrupt change in fluid properties at the interface can cause velocity instabilities to occur, the so called spurious currents. In order to avoid this, the interface can be given an artificial thickness to smooth the transition between phases. This thickness is selected to be a few element edge lengths. The thickness value is fixed, and it doesn't "smear" over time. It offers a trade-off between accuracy and simulation stability. Ideally, the thickness value is as close to zero as possible, as to give more accurate results, but for simulations where instabilities appear, increasing the value smooths the velocity fields providing a more stable, if slightly less accurate, result.

3.6.3 Heaviside Function

The Heaviside function is used as a step function to identify the fluid phases and smooth the fluid properties transition across the interface. When constructing the fluid interface, a vector normal to it is assigned to each interface node. When simulating a closed interface, such as a bubble or droplet, this normal vector is constructed to point outside the interface's surface. Using the normal vector to determine the sign, a signed distance $d(\mathbf{x})$ is calculated, measuring the distance from each fluid mesh node to the nearest interface node. This signed distance is used as input to the Heaviside function, which outputs a value of 1 for the inside of the interface and a value of 0 for the outside. If the fluid interface is given a finite thickness with the intent of smoothing out the sharp property transition across the interface, the Heaviside function outputs a value between 0 and 1 when evaluated inside the fluid interface. The Heaviside function utilized in this work is a smooth function detailed in [54], and is given by

$$H(\mathbf{x}) = \begin{cases} 1, & \text{if } d(\mathbf{x}) > \epsilon \\ 0, & \text{if } d(\mathbf{x}) < -\epsilon \\ 1 - 0.5\Big[1 + \dfrac{d(\mathbf{x})}{\epsilon} + \dfrac{1}{\pi} sin\big(\pi d(\mathbf{x})/\epsilon\big)\Big], & \text{otherwise} \end{cases} \tag{3.61}$$

where $d(\mathbf{x})$ is the signed distance from the evaluated fluid mesh node to the nearest interface node and ϵ is the parameter defining the interface thickness. The interface thickness is a numerical convenience, added to improve the simulation's stability by smoothing the fluid properties transition at the interface. It was found that a higher value is usually detrimental to the simulation's accuracy; therefore it's desirable to maintain its value low, usually in range of $1-2$ element sizes. The fluid phase's properties, $\rho(\mathbf{x})$, and $\mu(\mathbf{x})$, can be calculated using the Heaviside function, using the following formula

$$\rho(\mathbf{x}) = \rho_1 H(\mathbf{x}) + \rho_2[1 - H(\mathbf{x})] \qquad \mu(\mathbf{x}) = \mu_1 H(\mathbf{x}) + \rho_2[1 - H(\mathbf{x})] \tag{3.62}$$

where ρ_1, μ_1 represent the fluid specific mass and viscosity for the fluid inside the interface, and ρ_2 and μ_2 are the properties of the fluid outside the interface.

3.6.4 Interface Mesh Coupling

At the start of each time step, the interface mesh curvature is calculated. Since the interface mesh is not geometrically coupled to the fluid mesh, this curvature value needs to be assigned to the fluid mesh, in order to calculate surface tension. This assignment is accomplished by going through all fluid mesh nodes, identifying through a distance function the nearest interface node, and assigning the curvature value of the closest interface node to the corresponding fluid mesh node. The surface tension force is then calculated and the primary variables computed.

The interface mesh is treated based on a Lagrangian reference, and as the simulation progresses through time, its position needs constant updating to accommodate any movement. To achieve this, the velocity values for each interface node is necessary. A search is conducted on the fluid mesh to identify the specific fluid mesh element upon which each interface node resides. Following this determination, a linear interpolation of the velocity values of the fluid mesh element nodes is executed, and the resulting value is assigned to the corresponding interface node. This process is repeated for all interface nodes, and then their positions are updated by

$$\mathbf{x_i^{n+1}} = \mathbf{x_i^n} + \mathbf{v_i}\Delta t \tag{3.63}$$

where $\mathbf{x_i^{n+1}}$ is the new interface node position, $\mathbf{x_i^n}$ is the previous time step interface node position, $\mathbf{v_i}$ is the discreet velocity assigned to the interface node and Δt is the time step. On simulations that present instabilities due to spurious velocity fluctuations, each interface node position is updated based on its assigned velocity and the velocity of their neighbor nodes. Therefore the following equation can be used instead of the previous one:

$$\mathbf{x_i^{n+1}} = \mathbf{x_i^n} + \left(0.5\mathbf{v_i} + \frac{0.5}{n}\sum_{i=n}\mathbf{v_{neighbor}}\right)\Delta t \tag{3.64}$$

where n is the number of neighbors an interface node has, which are the nodes present in elements the node belongs to, or the nodes resulting from the umbrella operator applied to the analyzed node.

3.7 Matrix Representation

Equations 3.37 through 3.39 and equation 3.59 can be written as

$$\begin{aligned} &m\rho\dot{u} - q_x p + \frac{1}{Re}\mu(2k_{xx} + k_{yy} + k_{zz} + k_{yx} + k_{zx})u \\ &- \frac{1}{Fr^2}\rho m g_x - \frac{\nabla H}{We}(c_{xx} + c_{yy})x = 0 \end{aligned} \tag{3.65}$$

$$\begin{aligned} &m\rho\dot{v} - q_y p + \frac{1}{Re}\mu(k_{xx} + 2k_{yy} + k_{zz} + k_{xy} + k_{zy})v \\ &- \frac{1}{Fr^2}\rho m g_y - \frac{\nabla H}{We}(c_{xx} + c_{yy})y = 0 \end{aligned} \tag{3.66}$$

$$\begin{aligned} &m\rho\dot{w} - q_z p + \frac{1}{Re}\mu(k_{xx} + k_{yy} + 2k_{zz} + k_{xz} + k_{yz})w \\ &- \frac{1}{Fr^2}\rho m g_z - \frac{\nabla H}{We}(c_{xx} + c_{yy})z = 0 \end{aligned} \tag{3.67}$$

and

$$d_x u + d_y v + d_z w = 0 \tag{3.68}$$

These matrices encompass a single element, and to solve the equations for the velocity and pressure fields over the whole domain, it is necessary to solve for all elements simultaneously. To this end, the assembly process is necessary, where the local element matrices are added to a global matrix producing a linear system of equations. This is represented as follows

$$\begin{bmatrix} A & Q \\ \\ D & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u^{n+1}} \\ \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} B \\ \\ 0 \end{bmatrix} \tag{3.69}$$

where the terms A, G, and D are sub-matrices, and B is a vector given bellow by

$$A = \begin{bmatrix} \frac{\mu}{Re} 2K_{xx} + \frac{M}{\Delta t} & \frac{\mu}{Re} K_{xy} & \frac{\mu}{Re} K_{yz} \\ \frac{\mu}{Re} K_{xy} & \frac{\mu}{Re} 2K_{yy} + \frac{M}{\Delta t} & \frac{\mu}{Re} K_{yz} \\ \frac{\mu}{Re} K_{xz} & \frac{\mu}{Re} K_{yz} & \frac{\mu}{Re} 2K_{zz} + \frac{M}{\Delta t} \end{bmatrix} \tag{3.70}$$

$$Q = \begin{bmatrix} Q_x \\ \\ Q_y \\ \\ Q_z \end{bmatrix} \tag{3.71}$$

$$D = \begin{bmatrix} D_x & D_y & D_z \end{bmatrix} \tag{3.72}$$

and

$$B = \begin{bmatrix} \frac{M}{\Delta t} u^n + \frac{1}{Fr^2} \rho g_x + \frac{1}{We} f_x \\ \frac{M}{\Delta t} v^n + \frac{1}{Fr^2} \rho g_y + \frac{1}{We} f_y \\ \frac{M}{\Delta t} w^n + \frac{1}{Fr^2} \rho g_z + \frac{1}{We} f_z \end{bmatrix} \tag{3.73}$$

The choice for the matrix layout exhibited in 3.69, coupled with equation 3.41, leads to the matrix being symmetric, but not positive definite, which impacts the options available to solve it.

3.8 The Semi-Lagrangian Method

The material derivative in the equation of conservation of momentum contains a non-linear term $\mathbf{v} \cdot \nabla \mathbf{v}$, the convection term. When this term is added to the finite element discretization through the Galerkin method, the problem's solution might contain spurious oscillations depending on the simulation's Reynolds number and chosen time step.

These spurious oscillations can be avoided in a number of ways. For example, the Streamline Upwind Petrov-Galerkin (SUPG) achieves stabilization through the use of different weight functions when developing the weak form, in a way that balances the negative diffusion added by the central difference approximation. The Characteristic Galerkin procedures split the equation into a convective and diffusive equation, linking them through the boundary and initial conditions. Another way to deal with instability due to the convection term is the Characteristic Based Split scheme, which removes the pressure term from the momentum equation, allowing for different interpolation functions. The details of these procedures can be read in [51] and [55].

One convenient way to discretize the advection term is through the semi-Lagrangian method. it is a method that first encountered use on weather prediction [56], and offers simpler implementation compared to other options, preserves the symmetry of matrices and presents itself unconditionally stable for large time steps.

A purely Lagrangian advection scheme would follow a set of particles over the simulation time, but the particles' destination might not offer regularly spaced or convenient positions. On a finite element analysis, it is desirable to know the quantities at fixed positions, the mesh nodes. The semi-Lagrangian scheme achieves this goal by changing the set of particles at each time step, following the particles that will occupy the mesh nodes' positions at each time increment.

The particle set chosen at each time step is the set that after advection will arrive exactly at the mesh nodes. Thus at each time step, one needs to search for the points in the domain which will be transported to the mesh nodes position after a time interval Δt. Those are labeled departure points.

The material derivative of a given scalar function Ψ is given by

$$\frac{D\Psi}{Dt} = \frac{\partial \Psi}{\partial t} + u\frac{\partial \Psi}{\partial x} + v\frac{\partial \Psi}{\partial y} + w\frac{\partial \Psi}{\partial z} \tag{3.74}$$

This operator can be discretized by a first-order approximation given by

$$\frac{D\Psi}{Dt} = \frac{\Psi^{n+1} - \Psi^n_d}{\Delta t} \tag{3.75}$$

where Ψ_d is the value of Ψ at the departure point. The departure points are found by calculating

$$\mathbf{x}_d = \mathbf{x}^n - \mathbf{v}\Delta t \tag{3.76}$$

for each mesh node, where $\mathbf{x}_d$ is the departure point, $\mathbf{x}^n$ is the mesh node position and $\mathbf{v}$ the velocity vector at point $\mathbf{x}$. Once the departure point set is calculated, the scalar Ψ_d is assigned from interpolation of the known nodal values of Ψ.

3.9 Element Shape Functions

The element shape functions define how the field variable is approximated over the finite element, interpolating the field variable's value at the element's nodes. These functions are required to have:
1) Continuity over the problem domain up to one less than the highest derivative present in the weak form.
2) Linear independence, that is, no function can be obtained through a linear combination of other functions.
3) Completeness, all terms led by a constant must be present up to the highest order term. This requirement can be loosened if the lacking terms are "symmetric".
4) Equipresence, all the shape functions must have the same amount of terms that are function of x, y, and z.

The polynomial function is widely used, due to its ease of differentiation, the fact that it can be integrated numerically with exact results, and have its order increased to improve the accuracy of the results.

In this section, the Mini Element used in the discretization of the Navier-Stokes is described, along with the linear triangle element used in the discretization of the Laplace-Beltrami operator.

The variational formulation for the incompressible Navier-Stokes equations results in a saddle point problem. To achieve a solution for this problem, one needs to observe restrictions on the combination of interpolation functions for the velocity and pressure. To achieve a stable simulation for a mixed finite element method, and to guarantee the existence and uniqueness of the solution, the compatibility condition called Ladyzhenskaya-Babuska-Brezzi (LBB) must be satisfied, the details can be found in [57]. The LBB condition does not allow accurate results to be obtained when shape functions of the same order for the velocity and pressure are utilized in a standard finite element Galerkin discretization.

There are a number of methods to satisfy or circumvent the LBB condition, such as the pressure-stabilizing Petrov-Galerkin (PSPG) developed in [58] or the polynomial pressure projection method (PPPS) presented in [59]. One of the available methods involves the proper choice of finite element. Elements like the Taylor-Hood and Crouzeix and Raviart elements satisfy the condition. In this work, the mini element developed by Arnold, Brezzi, and Fortin [60] was utilized. The mini element interpolates the pressure with piecewise linear barycentric shape functions and interpolates the velocity fields with the piecewise linear barycentric shape functions enriched with a bubble function that vanishes on the element boundary. This bubble function is a product of the barycentric coordinates. Despite having a cubic (quartic for three-dimensional elements) shape function in the bubble function, the mini element exhibits linear convergence.

In the mini element, the linear shape functions are located at the element's corners, and the bubble function is located at the element's centroid node, therefore the velocity is assigned to five degrees of freedom per element, while the pressure has four, the element's corners. The bubble function is not a complete polynomial and therefore has a slower rate of convergence, however, it only adds one single extra degree of freedom. The name "mini" comes from the fact that the element possesses the minimal amount of degrees of freedom to satisfy the LBB condition.

3.9.1 Barycentric Coordinates

Before approaching the elements' shape function, it is important to describe the barycentric coordinates for a given triangle and tetrahedron, since those shape functions are based on the barycentric coordinates.

Triangle

In a given triangle, with corners i, j and k, for any point P arbitrarily located inside the triangle, the linear coordinates for the point P are

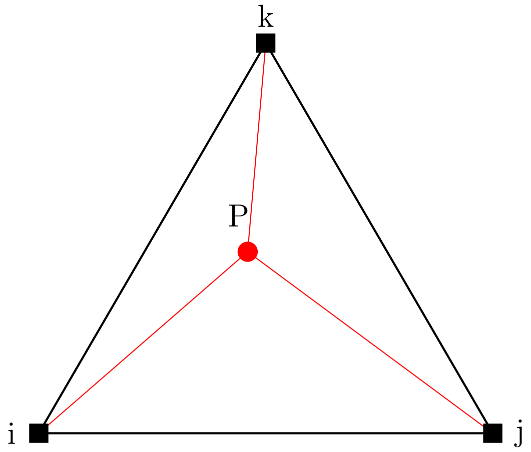


Figure 3.4: Area coordinates for a given generic triangle ijk, where P is an arbitrary point contained inside the triangle.

$$L_i^{tri} = \frac{A_i}{A} \qquad L_j^{tri} = \frac{A_j}{A} \qquad L_k^{tri} = \frac{A_k}{A} \tag{3.77}$$

where A is the triangle's total area, A_i is the area of the triangle created by points j, k and P, A_j is the triangle created by points i, k and P and A_k is the triangle formed by the points i, j and P. The sum of all linear coordinates must be

$$L_i^{tri} + L_j^{tri} + L_k^{tri} = 1 \tag{3.78}$$

The area of the triangle can be obtained by inputting the triangle's vertices' Cartesian coordinates into the following matrix

$$A = \frac{1}{2} det \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \tag{3.79}$$

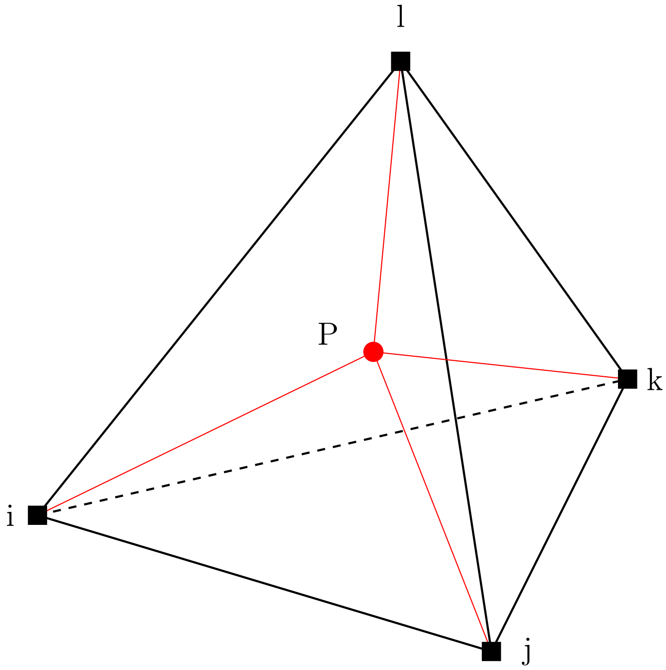


Figure 3.5: Volume coordinates for a given generic tetrahedron ijkl, where P is an arbitrary point contained inside the tetrahedron.

Tetrahedron

The barycentric coordinates for a tetrahedron are analogous to the triangle coordinates presented above. Given a tetrahedron with corners i, j, k and l, and a point P located at an arbitrary position inside the element, as given by fig. 3.5, the linear coordinates of point P are

$$L_i = \frac{V_i}{V} \qquad L_j = \frac{V_j}{V} \qquad L_k = \frac{V_k}{V} \qquad L_l = \frac{V_l}{V} \tag{3.80}$$

where V_i is the volume of the tetrahedron shaped by the points P, j, k, and l and so on. As long as point P is inside the element, the sum presented below is guaranteed

$$L_i + L_j + L_k + L_l = 1 \tag{3.81}$$

The volume of any tetrahedron can be found by inputting the point's Cartesian coordinates into

$$V = \frac{1}{6} det \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{bmatrix} \tag{3.82}$$

3.9.2 Linear Triangle Element

The triangle element has three shape functions, one assigned to each vertex. Its shape functions are quite simple, given by its barycentric coordinates

$$N_i = L_i^{tri} \qquad N_j = L_j^{tri} \qquad N_k = L_k^{tri} \tag{3.83}$$

3.9.3 Mini Element

The representation for the Mini element can be seen in fig. 3.6 and its shape functions are given by

$$\begin{aligned} &N_i = L_i - 64 L_1 L_2 L_3 L_4, \qquad for \quad i = 1, 2, 3, 4 \\ &N_5 = 256 L_1 L_2 L_3 L_4 \end{aligned} \tag{3.84}$$

where the indexes $i = 1, 2, 3, 4$ represent the element's corner nodes, and $i = 5$ represents the node at the element's centroid.

3.10 Finite Element Mesh

In this work, unstructured meshes were chosen for their greater flexibility in adapting to complex geometries.

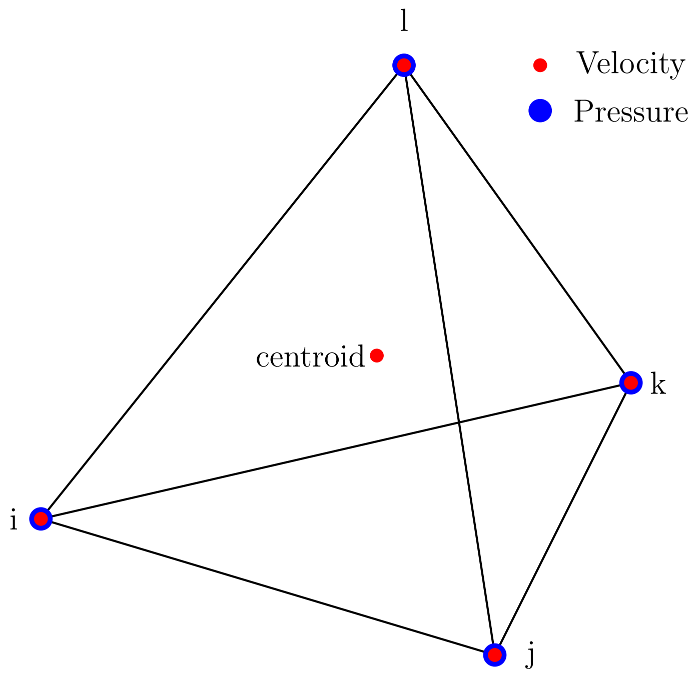


Figure 3.6: Mini element representation, indicating the nodes assigned to the velocity and pressure. The pressure is assigned to nodes on the tetrahedron's corners, and the velocity is assigned to the corners, plus the node located at the centroid. The integration on such coordinates can be done analytically, avoiding the costs of numerical integration.

Unstructured meshes are meshes created by a non-uniform node distribution and can be composed of irregularly shaped elements, including different element sizes. Unstructured meshes also support different element shapes, such as tetrahedrons and hexahedrons in the same mesh, as long as their interconnectivity is respected.

This non-uniform node distribution allows for many different geometries to be represented. Where a structured mesh might be limited to regular shapes, with straight boundaries, an unstructured mesh can adapt to irregular, changing, or curved domains, through the use of piecewise linear shapes. The use of unstructured meshes with the finite element method allows for great flexibility in problem domain geometries.

The element size in unstructured meshes can vary greatly, allowing for selective refinement of the problem domain, increasing refinement on areas where phenomena of interest happen or the field variables present steeper gradients. It also allows for a decrease in mesh resolution far from regions of interest to save on computational costs, such as memory and simulation run time.

While structured meshes can also have refinement of particular areas and differently sized elements, these are often much less flexible and often generate a greater number of elements compared to an unstructured mesh. A problem discretized by a mesh with more elements usually requires more computational time to be solved.

As a trade-off, due to the arbitrary connectivity between elements, unstructured meshes require more memory to store and produce matrices with a sparser structure, which can be slower to solve or restrict the choice of linear system solver.

Since an unstructured mesh is not constrained to regularly shaped elements, care must be taken so that elements present in the mesh are well-shaped. For this reason, the Delaunay tetrahedralization is utilized. The Delaunay tetrahedralization constructs the mesh based on the dual graph of the Voronoi Diagram. It is a popular choice for mesh construction and ensures the elements produced have good quality [61].

To compare different results produced by a problem, and to measure the refinement level of a given mesh, some type of element size metric must be utilized. To this end, the average edge length, h was selected. it is an average because since the mesh is unstructured, it cannot have a fixed edge length. The average edge length offers an easy to understand metric when compared to the domain length, as opposed to the average element volume, and it is the measure inputted directly into the mesh generator. The average edge length can be calculated by

$$h = \frac{\sum_{n}^{i} l_i}{n} \tag{3.85}$$

where l is the edge length and n is the number of edges contained in the mesh. In meshes with different refinement levels, multiple average edge lengths might be utilized, according to the selected refinement for each region of the problem's domain.

3.11 Time Step Calculation

When executing a transient problem simulation, the simulation progresses in time. The amount of time advanced is the time step. Ideally, this time step should be small enough that the simulation captures the transient phenomena one wishes to observe, such as the flow becoming fully developed, or the wake oscillations after a bluff body, but large enough so that the simulation does not take an excessive amount of time.

Another concern is that some discretization schemes for the Navier-Stokes equations contain terms that are conditionally stable. If the conditions imposed by those terms are not met, the simulation does not produce accurate results. The conditionally stable terms restrict the choice for the value of the time step increment. Generally, the terms treated implicitly are unconditionally stable, while the explicitly treated terms can be conditionally stable or unstable. More details about stability can be found in [62]

In most discretization schemes, the convection term requires a time step restriction. In this work, the material derivative is discretized explicitly by the semi-Lagrangian technique. The semi-Lagrangian is unconditionally stable, not presenting any restriction on the time step utilized. it is desirable, however, to maintain the time step under a certain limit, to preserve the method's accuracy. The semi-Lagrangian method has first-order accuracy in time, and the fluid's trajectory is considered a straight line. A larger time step increases the method's error. To reduce the concerns over the method's accuracy, the time step constraint for the semi-Lagrangian discretization of the convective term should be limited by:

$$\Delta t_c < \frac{h_{min}}{\mathbf{u}_{max}} \tag{3.86}$$

where h_{min} is the smallest element edge size of the fluid mesh, and $\mathbf{u}_{max}$ is the maximum velocity value, of any of the three velocity components, u, v, w

The other terms treated explicitly in this work are the gravity force and the surface tension force. The surface tension term adds a constraint to the simulation time step, as reported by [63] given by

$$\Delta t_{sf} < \sqrt{\frac{We\rho_{av}h^3}{2\pi}} \tag{3.87}$$

and the gravity force constrains the time step by the following equation, as denoted by [64],

$$\Delta t_g < \sqrt{\frac{1}{h_{min}}} \tag{3.88}$$

The time step that should be adopted when running a simulation, is the smallest of the listed values above, or

$$\Delta t = min\{\Delta t_g, \Delta t_{sf}, \Delta t_c\} \tag{3.89}$$

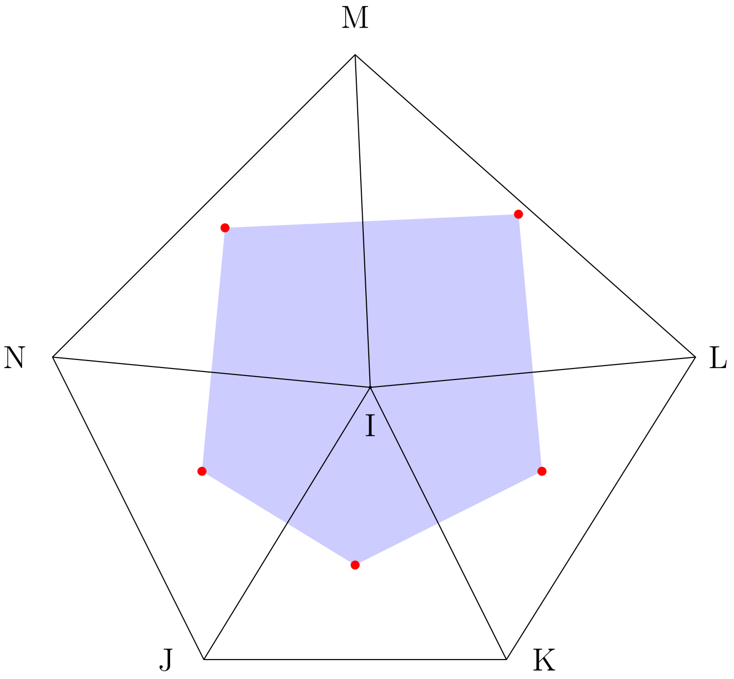


Figure 3.7: Voronoi area for a node I in a triangular surface mesh. The blue area bounds the Voronoi area for node I, belonging to the five triangular elements shown. The red dot represents each triangle's circumcenter.

3.12 Interface Element Area Calculation

To calculate the curvature at a given node it is necessary to assign a discrete area value to it. In a surface mesh composed of triangular elements, the nodes are the triangle's vertices. One way to calculate the area around a node is the Voronoi area, which is found by taking the midpoint of each triangle edge connected to a given node and joining it to the triangle's circumcenter. The resulting area is the Voronoi area and can be observed in fig. 3.7. An alternative to using the Voronoi area is performing the same process, but using the centroid instead of the circumcenter. This produces the barycentric area, but this method presented less accurate results when tested, compared to the Voronoi area.

A node belongs to multiple triangles in a mesh. To calculate the circumcenter area for a given node **I**, the first step is to pick a triangle to which the node belongs to and calculate the triangle's circumcenter coordinates

$$\mathbf{C} = \frac{a^2(b^2+c^2-a^2)\mathbf{I} + b^2(a^2+c^2-b^2)\mathbf{J} + c^2(a^2+b^2-c^2)\mathbf{K}}{a^2(b^2+c^2-a^2) + b^2(a^2+c^2-b^2) + c^2(a^2+b^2-c^2)} \tag{3.90}$$

where $\mathbf{C}$ is the triangle's circumcenter and $\mathbf{I}$, $\mathbf{J}$ and $\mathbf{K}$ are the nodes at the triangles vertices. One can define the following vectors

$$\begin{aligned}
\mathbf{t_1} &= \frac{\mathbf{J}-\mathbf{I}}{2} \\
\mathbf{t_2} &= \frac{\mathbf{K}-\mathbf{I}}{2} \\
\mathbf{m_1} &= \mathbf{C} - \frac{\mathbf{J}+\mathbf{I}}{2} \\
\mathbf{m_2} &= \mathbf{C} - \frac{\mathbf{K}+\mathbf{I}}{2}
\end{aligned} \tag{3.91}$$

and calculate the circumcenter area by

$$A_{circ} = \frac{1}{2}(\mathbf{t_1} \times \mathbf{m_1} + \mathbf{t_2} \times \mathbf{m_2}) \tag{3.92}$$

The Voronoi area of a given node is obtained by repeating this process for each triangle it belongs to, and summing the area values.

3.12.1 Obtuse Triangle Area Correction

The mean curvature calculated with $A_{Voronoi}$ proved to be reasonably accurate as long as no obtuse triangles were found on the interface mesh. A single obtuse triangle can skew the mean curvature values of the nodes belonging to it by a very large margin, and refining the mesh shows no improvement at all. In fact, refining the interface meshes can present even worse mean curvature values, if the refinement process includes more obtuse angles in the elements.

For a right or acute triangle, the circumcentric areas of the three vertices add up to the triangle's area. For obtuse triangles, this is not the case, and the sum of all circumcentric areas is greater than the triangle's area. It seems this is what induces the error in mean curvature.

In [65] the use of barycentric areas for obtuse triangles is proposed, and [66] proposes to use 1/2 of the triangle's area for the vertex with the obtuse angle and 1/4 of the triangle's area for the other two vertices. However, these methods do not show significant improvement, and depending on the mesh, can worsen the mean

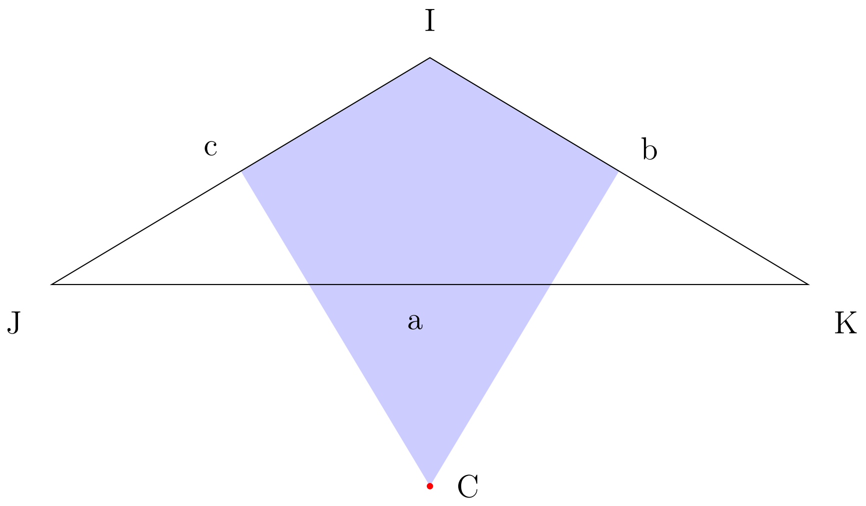


Figure 3.8: Obtuse Triangle IJK and its circumcenter C, marked as a red dot outside the triangle's boundaries. The Voronoi area for this triangle exceeds the triangle's total area when calculated for all three vertices. The area calculated using the edges connected to the obtuse angle causes the distortion.

curvature error.

In this work, we used the following correction, which proved to be both accurate and straightforward to implement. An obtuse triangle is given, with vertices **I**, **J**, **K** and circumcenter **C**. In this triangle, the vertex **I** has the obtuse angle, as shown in fig. 3.8. The areas are calculated as follows

$$\begin{aligned}
A_I &= A_{circ} \\
A_J &= \frac{c}{b+c}(area - A_I) \\
A_K &= \frac{b}{b+c}(area - A_I)
\end{aligned} \tag{3.93}$$

where *area* is the triangle's area, b the edge length opposite to vertex J and c the edge length opposite to vertex K. it is important to note that A_J and A_K can be negative.

3.13 Numerical Tools

All meshes utilized in this work, both for the fluid and fluid interface were generated using the free mesh generator Gmsh [67].

The Finite Element Method code, including the mesh reading and file outputting, was developed by the author, using C++ language, based on the object-oriented paradigm. The linear algebra operations and the solution for the linear systems of equations were generated using the library EIGEN.

The code was compiled with the GNU Compiler Collection, on both Linux and Windows machines.

Most post-processing and visualization is handled by Paraview

Eigen, gmsh, and paraview can be found, respectively, at:

- https://eigen.tuxfamily.org/index.php?title=Main_Page
- https://gmsh.info/
- https://www.paraview.org/

3.13.1 Remeshing of the Unstructured Interface Mesh

The interface mesh nodes are advected by the velocity fields proceeding from the finite element solution of the fluid mesh. This velocity fields that promote the advection are not uniform, and over simulation time, the interface mesh will have distorted elements, lowering their quality and eventually becoming unusable. In order to avoid this, a remeshing procedure that maintains the mesh quality while preserving the geometry and underlying curvature is necessary.

To this end, the PMP library is utilized [68]. It implements the remeshing algorithm described on [69] and [70]. It reconstructs the mesh using the Adaptive Isotropic Remeshing technique, which uses a combination of edge collapses and splits, edge flipping and moving edge positions. The PMP library can be found at:

- http://www.pmp-library.org/

The remeshing algorithm targets an average edge size equal to the starting average edge size, therefore element size is roughly the same as the initial element size. The mesh produced however presents an slight loss of volume, which is corrected by moving the nodes in their normal direction until the volume matches the initial volume.

3.13.2 Linear System Solver

The linear system of equations presented in 3.69 is symmetric, but not positive definite, due to the zeros in its main diagonal. This fact restricts the options for the solver of a linear system of equations. For this work, both the minimum residuals method [71] and the projection method [72] were chosen to solve the system of equations.

The minimum residuals can be applied directly, using the linear algebra library available, while the projection method requires a decomposition of the linear system matrix. The selected decomposition in this work is presented bellow

$$\begin{bmatrix} A & Q \\ D & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ D & (E + DA^{-1}Q) \end{bmatrix} \begin{bmatrix} I & A^{-1}Q \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{v}^{n+1} \\ p^{n+1} \end{bmatrix} \tag{3.94}$$

where the first matrix on the right side is called the L matrix and the second matrix on the right side is called the U matrix. Based on this decomposition, one can solve

$$A\mathbf{v}_{trial} = \frac{M}{\Delta t}\mathbf{v}^n \tag{3.95}$$

to find a "trial velocity" value, $\mathbf{v}_{trial}$. This value then is replaced into the equation below to find the pressure values

$$(E + DA^{-1}Q)\mathbf{p}^{n+1} = D\mathbf{v}_{trial} \tag{3.96}$$

and finally replace the trial velocity and pressure values into the following equation to find the velocity values

$$\mathbf{v}^{n+1} = \mathbf{v}_{trial} + A^{-1}Q\mathbf{p}^{n+1} \tag{3.97}$$

The evaluation of A^{-1} is a costly operation, turning the execution of all but the most simple and unrefined problems unfeasible. To avoid this issue, an approximation is made to the value of A^{-1}. There are some options to this end, but the simplest one, adopted in this work is the construction of a matrix A_{lumped}. This is done by summing the values of each row in the matrix and adding it to its respective main diagonal position. The end result is a diagonal matrix, which is easily invertible.

Chapter 4

Numerical Results

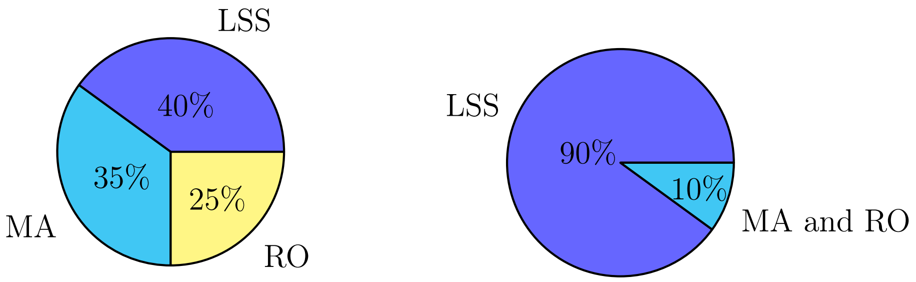
To assert the accuracy of the results offered by the methodology and its numerical implementation, code verification is necessary. The code's results can be compared against well-known analytical or numerical test cases in increasing order of complexity.

This chapter covers the simulation of Hagen-Poiseuille flow and compares it against the exact solution for steady state conditions, evaluating the accuracy of the proposed methodology for single-phase flows. The mean curvature values obtained through the finite element method discretization of the Laplace-Beltrami operator are also presented, and their accuracy is assessed. Next, the simulations of several two-phase flow problems are presented, which are compared to existing data to evaluate the accuracy of the proposed methodology for two-phase flows.

The majority of the tests executed were solved through the projection method, except for the Poiseuille test and the static droplet test, which were solved directly using the minimum residuals method. This is because, due to the approximations made in the projection method, the minimum residuals proved more accurate for very low (approximately smaller than 5) Reynolds numbers.

The Poiseuille, Static Droplet and Oscillating Droplet tests were executed on a machine with an Intel Xeon Gold 6348 processor, $2.6GHz$, using up to 6 cores. The remaining tests were executed on a machine with an Intel Xeon Silver 4316, $2.3GHz$, using 2 cores. The code uses approximately 5 gigabytes of RAM per million fluid mesh nodes on average, with occasional spikes reaching up to 10 gigabytes on some operations. It is important to note that the code only uses multiple cores on matrix multiplications and in the solution of the linear system of equations.

A rough estimation of the time taken by each step when executing the simulation is displayed in fig . 4.1a. The category presented as "Remaining Operations" include the semi-Lagrangian search, the interface remeshing procedure, the calculation of surface tension and gravity forces, the output writing. The quantities displayed are not precise, and the semi-Lagrangian search can sometimes take much more time,



(a) Projection method.

(b) Minimum residuals method.

Figure 4.1: Percentage of time spent by simulation procedures for each linear system solver. LSS stands for "Linear system solver", MA represents "Matrix Assembly" and RO "Remaining Operations". Using the minimum residuals method is overall slower than using the projection method.

depending on the problem. When using the minimum residuals method, the solution of the linear system of equations is much slower than the projection method, taking up most of simulation time. The time taken by some of the simulations executed can be observed in table 4.1.

Table 4.1: Time spent by simulations in hours.

Simulation	Solver	No. Nodes	No. Iterations	Time (hours)
Static Droplet	Min. Residuals	1132910	1492	105.2
Oscillating Droplet	Projection	594381	400	3.2
Rising Bubble	Projection	1012381	400	21.5
Taylor (sucrose)	Projection	709579	1250	32.4

4.1 Single-Phase Simulation

4.1.1 Hagen-Poiseuille

The Hagen-Poiseuille flow is a standard benchmark for fluid flow simulation. The exact solution is readily obtained by simplifying the Navier-Stokes equations, assuming a fully developed, steady-state flow, with the only non-null velocity component parallel to the tube, which results in the following equation

$$\nabla p = \nu \nabla^2 \mathbf{u} \tag{4.1}$$

where the solution, which takes a parabolic shape, can be written as

$$u = 2u_{av}\left(1 - \frac{r^2}{r^2_{max}}\right) \tag{4.2}$$

here, u is the velocity in the flow direction, u_{av} is the average velocity in the flow direction, which on a real flow can be obtained by the mass flow rate, r is the radius coordinate, starting from the tube center, and r_{max} is the tube's radius.

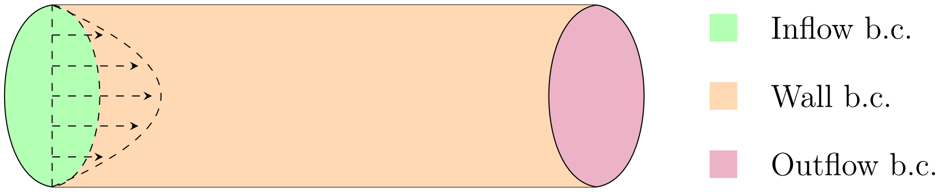


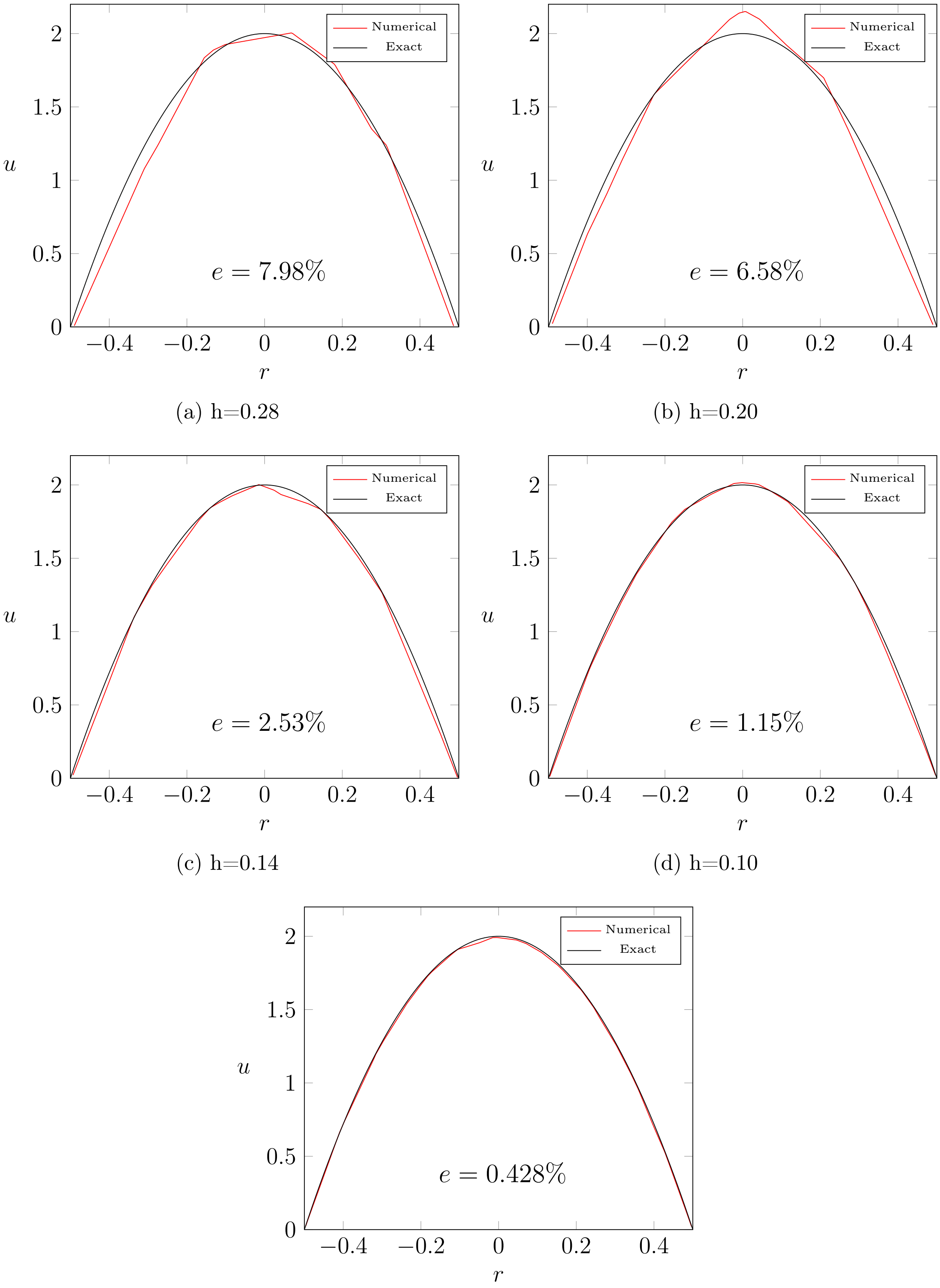
Figure 4.2: Finite element simulation parameters for the Hagen-Poiseuille test case. A cylindrical tube section is displayed, with the inflow region boundary conditions set as $v = w = 0$ and u set to the results of a parabolic curve, $u = 2-\frac{2r^2}{025}$. The outflow boundary conditions were $p = 0$, and at the tube section's walls, $u = v = w = 0$ was set.

The analytical velocity profile, assuming an average velocity of $u_{av} = 1$, and a diameter $d = 1$ is plotted in black in fig. 4.3.

The simulation was set as shown in fig. 4.2. The cylindrical tube section has a diameter of $d = 1$, and length $l = 3$. At the inflow, as boundary conditions, the velocity values were set to $u = 1$, $v = w = 0$. At the tube walls, no-slip boundary conditions were assigned, $u = v = w = 0$, and at the outlet, outflow boundary conditions were assigned, $p = 0$.

A convergence test was performed for this example, with five finite element meshes, with element edge size $h_f = 0.28$ (1299 nodes, 989 elements), $h_f = 0.2$ (2537 nodes, 1971 elements), $h_f = 0.14$ (7097 nodes, 5691 elements), $h_f = 0.1$ (18520 nodes, 15171 elements) and $h_f = 0.071$ (50574 nodes, 42068 elements), each h a factor of $\sqrt{2}$ of the previous one. The simulation parameters for all meshes were time step $\Delta t = 0.01$ and the Reynolds number $Re = 1$. The simulation was executed over 20 time steps, achieving steady state. The simulation's horizontal velocity profile can be observed in fig. 4.3, plotted against the analytical result.

In fig. 4.4 the velocity in the x direction and pressure fields, obtained from the most accurate mesh, $h_f = 0.071$ are displayed. In tab. 4.2, the error for each mesh is presented, and in fig 4.5 a log-log comparison to a linear function is displayed. The error was calculated by comparing the velocity values in the flow direction for all



(a) h=0.28

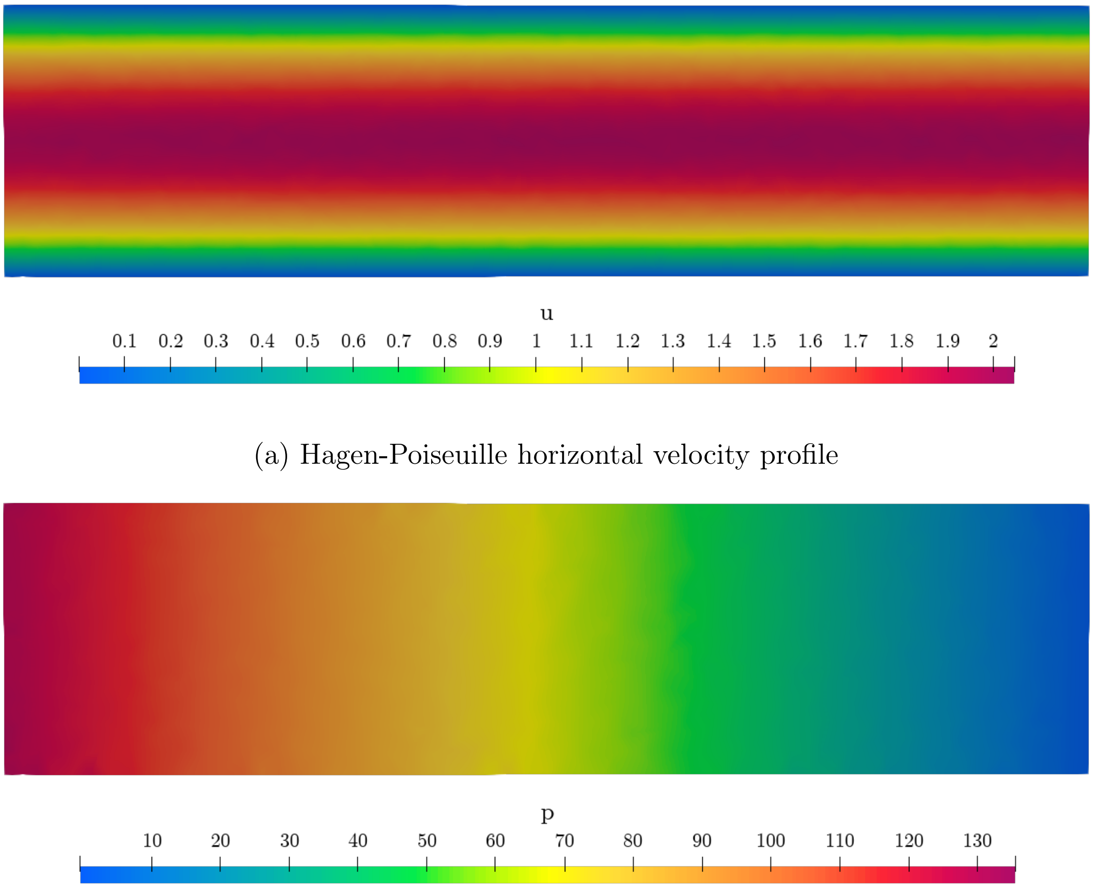
(b) h=0.20

(c) h=0.14

(d) h=0.10

(e) h=0.071

Figure 4.3: Hagen-Poiseuille flow velocity profile. The numerical results are plotted in red, while the exact results are plotted in black, for the meshes constructed by (a) 1299, (b) 2537, (c) 7097, (d) 18520, and (e) 50574 nodes, respectively.



(a) Hagen-Poiseuille horizontal velocity profile

(b) Hagen-Poiseuille pressure profile

Figure 4.4: Velocity profile for the x direction, and pressure profile for the Hagen-Poiseuille test case, with average edge length $h_f = 0.071$ (50574 nodes, 42068 elements). Both pictures show the plane obtained when the cylinder is cut in half.

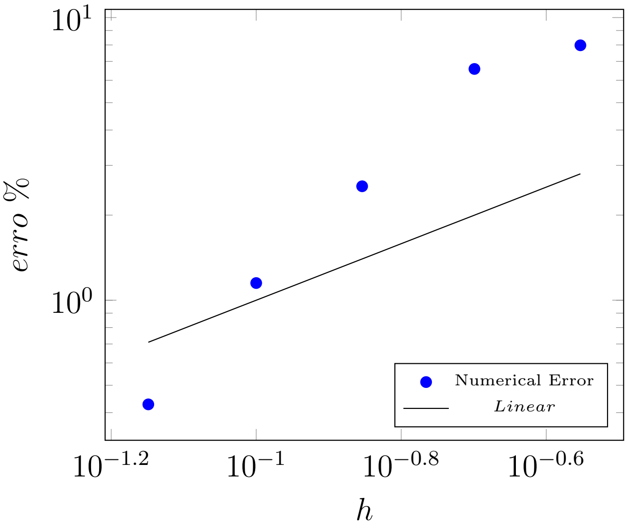


Figure 4.5: Calculated error for horizontal velocity component in the Poiseuille test case in comparison with a linear function ax. The simulation details for these tests are detailed in Tab. 4.2.

the simulation nodes to the analytical values, obtained from 4.2. The error formula utilized is

$$\|e\|_2 = \sqrt{\left(\frac{\sum(u_n - u_a)^2}{\sum u_a^2}\right)} \tag{4.3}$$

where e is the relative error, u_n is the nodal value for the velocity in the x direction and u_a is the analytical value for the same nodal position.

Table 4.2: Error values for the horizontal (along the tube's axis) velocity, for the Hagen-Poiseuille finite element test cases

Edge Length (h)	Number of Nodes	e
0.28	1299	7.98%
0.2	2537	6.58%
0.14	7097	2.53%
0.1	18520	1.15%
0.071	50574	0.428%

4.1.2 Single-phase Flow in Several Cross Sections

In this section, the results of the simulation of microchannels with multiple cross-sections are presented. Microchannels are frequently used in heat transfer applications, and they are crucial to understand their fluid flow characteristics.

The results for circular, triangular, square, rectangular, and hexagonal cross sections are presented. The circular cross-section presented in this section differs from the one shown in section 4.1 by its inflow boundary conditions. The simulation configurations are displayed in fig. 4.6. All cross sections have the same hydraulic diameter $d_h = 1$, calculated by inputting $d_h = 1$ in the following equation and solving for the edge size

$$d_h = \frac{4A_c}{p} \tag{4.4}$$

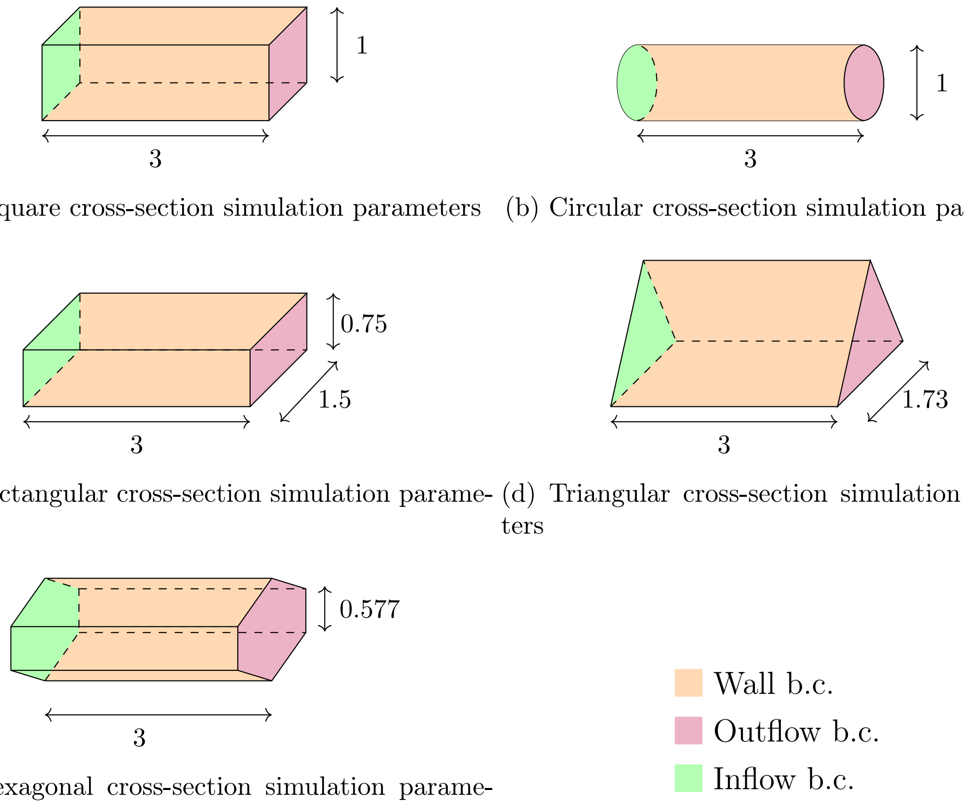
where A_c is the cross-sectional area, and p is the perimeter, both of them function of edge size.

All fluid domains for the five tests were set up in the same way. At the entrance, velocity is set to $u = 1$ and $v = w = 0$. At the walls of the channels, no-slip boundary conditions were assigned, $u = v = w = 0$, and at the channel's exit, outflow boundary conditions are used, $p = 0$. The channel length is the same for all cross sections, $length = 3$. Reynolds number $Re = 1$ is chosen, as it represents a typical Reynolds number for a microchannel filled with refrigerant fluid. The average edge length used is the same for all cases, $h_f = 0.04$, the mesh with circular cross-section has 197337 nodes and 166584 elements, the mesh with triangular cross-section has 329569 nodes and 278351 elements, the mesh with square cross-section has 251361 and 212216 elements, the mesh with rectangular cross-section has 285330 nodes and 240883 elements and the mesh with hexagonal cross sections has 216659 nodes and 182789 elements.

In tab. 4.3, the pressure drop and channel entry length are presented. The pressure drop was evaluated by calculating the average of all pressure values of the nodes at the inflow.

When the flow is fully developed, the transversal velocities, v and w are null. Due to numerical and discretization errors, the velocity values for the transversal velocities in the present simulation are not actually null. Consequently, the criteria used for evaluating the entry length was checking at which length the transversal velocities assumed values smaller than 1% the maximum velocity in the x direction. The channel entry length is also presented in tab. 4.3.

In fig. 4.7, the horizontal and vertical velocity profiles are displayed. The velocity profile for the circular section is the same as the one described in chapter 4 for the Hagen-Poiseuille case. There is a distinct decrease in velocity close to the triangle's



(a) Square cross-section simulation parameters

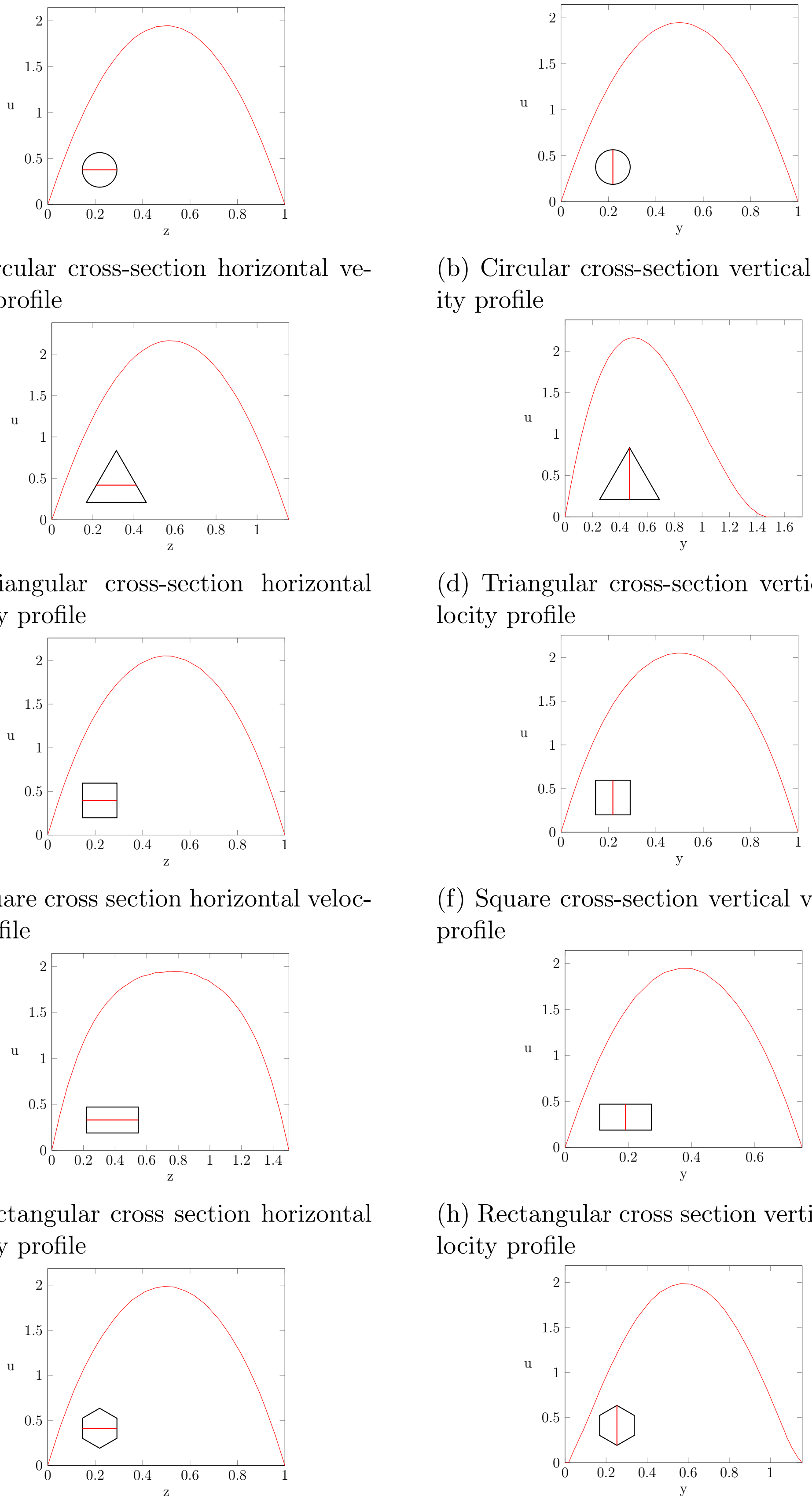
(b) Circular cross-section simulation parameters

(c) Rectangular cross-section simulation parameters

(d) Triangular cross-section simulation parameters

(e) Hexagonal cross-section simulation parameters

Figure 4.6: Microchannel simulation parameters for all cross-sections. All cross-sections have the same hydraulic diameter $d_h = 1$ and channel length $l = 3$. All polygonal cross-sections are regular, except for the rectangular one. All simulations were executed with $\mathit{Re} = 1$



(a) Circular cross-section horizontal velocity profile

(b) Circular cross-section vertical velocity profile

(c) Triangular cross-section horizontal velocity profile

(d) Triangular cross-section vertical velocity profile

(e) Square cross section horizontal velocity profile

(f) Square cross-section vertical velocity profile

(g) Rectangular cross section horizontal velocity profile

(h) Rectangular cross section vertical velocity profile

(i) Hexagonal cross section horizontal velocity profile

(j) Hexagonal cross section vertical velocity profile

Figure 4.7: Horizontal and vertical velocity profiles for the five different cross sections. The red line marks the position where the velocity profile was evaluated.

Table 4.3: Pressure drop and entry length for multiple channel cross sections

Cross Section	Pressure Drop	Entry Length
Circular	114.6	0.57
Triangular	97.5	0.98
Square	106.0	0.66
Rectangular	113.2	0.94
Hexagonal	111.9	0.61

vertices in the triangular cross-section. The hexagonal and square cross-section velocity profiles are very close to the circular one.

In fig. 4.8 contour lines describing the velocity in the x direction can be observed, for all cross sections. The velocity values for each contour line, from outside to inside, are $u = 0.01$, $u = 0.4$, $u = 0.8$, $u = 1.2$, $u = 1.6$, and $u = 1.9$. The square cross-section presents a steeper velocity gradient close to the walls, but a less pronounced gradient at the center. This suggests this section might perform differently from the circular one when simulating two-phase flow. The more even velocity values at the center might provide a more stable numerical simulation.

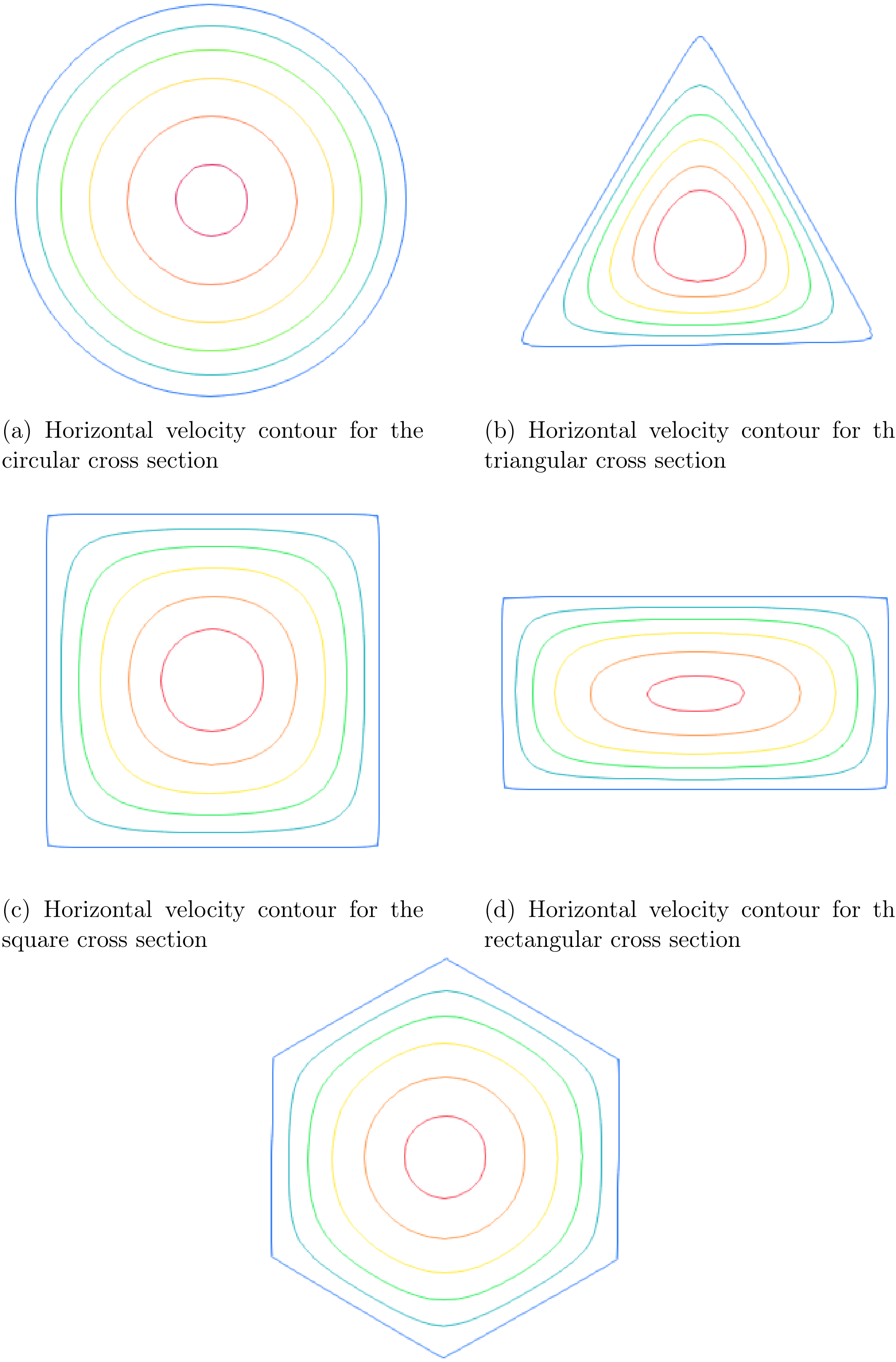
The difference in velocity contours implies that the geometry of the channel influences the flow significantly, and care must be taken when comparing the results to experimental data. Even if the fluid properties are the same, the geometric parameters might differ, causing differences in bubble shapes for example.

4.2 Mean Curvature Evaluation in Arbitrary Geometries

The fluid interface mean curvature is a crucial component in the calculation of the surface tension force. The precision of the mean curvature calculation influences directly the two-phase simulation's accuracy and stability, therefore, comparing the values obtained numerically to known values is a procedure to further guarantee the validity of the two-phase flow simulation. This comparison is presented in this chapter.

4.2.1 Sphere

For any given sphere, the mean curvature at any point is invariant, given by $\kappa = 1/2r$, where r is the radius. For a sphere of radius $r = 0.5$, used in all sphere tests, the mean curvature is equal to $\kappa = 4$. In fig. 4.9 one of the meshes used for the sphere's mean curvature calculation can be observed. The mesh presented has



(a) Horizontal velocity contour for the circular cross section

(b) Horizontal velocity contour for the triangular cross section

(c) Horizontal velocity contour for the square cross section

(d) Horizontal velocity contour for the rectangular cross section

(e) Horizontal velocity contour for the hexagonal cross section

Figure 4.8: Horizontal velocity contour plot at the outflow. From outside to inside, the velocities represented are $u = 0.01, u = 0.4, u = 0.8, u = 1.2, u = 1.6$, and $u = 1.9$.

an average edge length of $h_i = 0.085$, and is constructed with 405 nodes and 806 triangular surface elements.

Table 4.4: Mean curvature (κ) values for the sphere, for meshes of different average edge lengths. On the table, the error, maximum error, error after area correction, and maximum error after area correction are displayed.

Average Edge Length (h)	e	e_{max}	e_{ac}	e_{acmax}
0.16	8.02%	1.79	1.76%	0.620
0.12	4.33%	1.84	1.17%	0.570
0.085	3.41%	1.75	0.762%	0.420
0.06	0.477%	0.404	0.206%	0.177
0.03	1.24%	1.49	0.407%	0.655
0.02	0.795%	1.67	0.284%	0.396

In tab. 4.4, the mean curvature error for several meshes is displayed, along with the maximum error found in each mesh. The meshes have an average edge length $h_i = 0.16$ (159 nodes, 313), $h_i = 0.12$ (317 nodes, 630 elements), $h_i = 0.085$ (625 nodes, 1246 elements), $h_i = 0.06$ (1141 nodes, 2277 elements), $h_i = 0.03$ (4308 nodes, 8612 elements) and $h_i = 0.02$ (9500 nodes, 18996 elements). One can note that refining the mesh reduces both the error and the maximum error, as expected.

On the same table are listed the values for mean curvature error and maximum error after the area correction proposed in section 3.12.1 is applied. The presented values show slightly improved accuracy and a markedly reduced maximum error.

The error was calculated according to the following equations

$$e = \sqrt{\left(\frac{\sum(\kappa_n - \kappa_a)^2}{\sum \kappa_a^2}\right)} \qquad e_{max} = \frac{|\kappa_{max} - \kappa_a|}{\kappa_a} \tag{4.5}$$

4.2.2 Cylinder

The cylinder in fig. 4.10 presents a radius $r = 0.5$ and a length of $l = 2$. Its mean curvature values are $\kappa = 2$ for the curved surface and $\kappa = 0$ for the cylinder's bases.

In tab. 4.5, the error and maximum error, discarding the nodes at the edges can be observed. The error and maximum error after area correction can also be observed in the same table. The meshes presented have an average edge length $h_i = 0.16$ (251 nodes and 498 elements), $h_i = 0.12$ (431 nodes and 858 elements), $h_i = 0.085$ (798 nodes and 1792 elements), $h_i = 0.06$ (1600 nodes and 3196 elements), $h_i = 0.03$ (6206 nodes and 12408 elements), $h_i = 0.02$ (13853 nodes and 27702 elements). As the mesh is refined, the error and maximum error are reduced. At the top

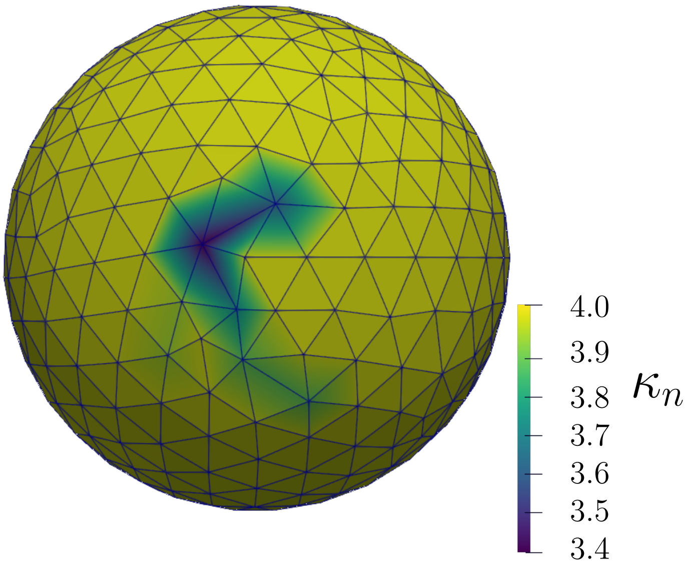


Figure 4.9: The nodal values for mean curvature on a sphere of radius $r = 0.5$. The mesh has an average edge length of $h_i = 0.085$ (405 nodes, 806 elements).

and bottom flat surfaces, the mean curvature values were of the order of 10^{-14}, which can be considered 0 for floating-point numbers in double precision. At the cylinder's edges, due to the non-smooth change in geometry, the curvature cannot be calculated properly, as it should increase towards infinity. The errors were calculated using equations 4.5.

Table 4.5: Mean curvature (κ) values for the cylinder's body, for meshes of different average edge lengths meshes. On the table, the error, maximum error, error after area correction, and maximum error after area correction are displayed.

Average Edge Length (h)	e	e_{max}	e_{ac}	$e_{ac_{m}ax}$
0.16	2.40%	0.157	2.40%	0.157
0.12	1.45%	0.140	1.45%	0.140
0.085	2.13%	0.130	2.13%	0.130
0.06	1.85%	0.207	1.88%	0.222
0.03	0.786%	0.210	0.781%	0.210
0.02	1.03%	0.219	1.02%	0.192

4.2.3 Torus

The torus is an important case to test curvature values because it can present negative curvature values. it is important to ensure the code is calculating the curvature sign correctly. In fig. 4.11 a Torus with major radius $R = 0.5$ and minor radius $r = 0.4$ is displayed. This configuration displays negative curvature values along the surface close to the center, as can be observed in the figure.

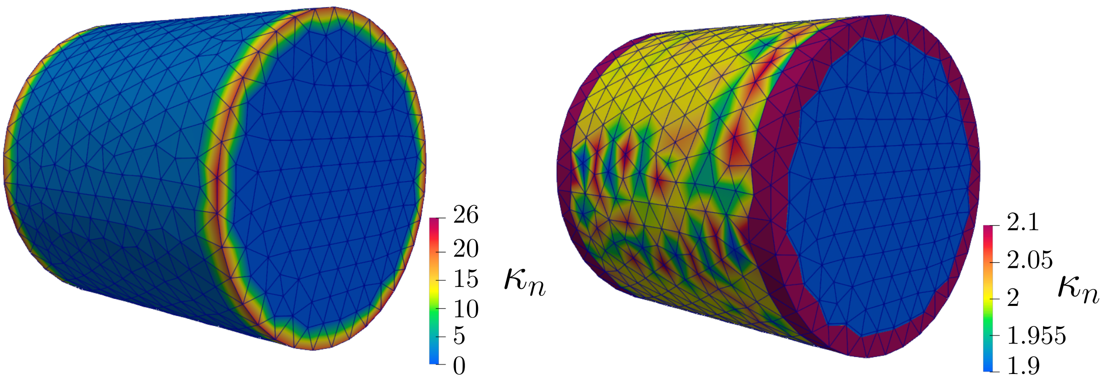


Figure 4.10: The nodal values for mean curvature on a cylinder, with average edge length $h_i = 0.085$ (798 nodes and 1792 elements). a) Mesh pictured on a scale presenting the maximum and minimum mean curvature values, showing the increased mean curvature values at the cylinder's edges. b) Mesh presented with custom scale, between $\kappa = 1.9$ and $\kappa = 2.1$, to reveal different nodal values inside the curved surface.

Table 4.6: Mean curvature (κ) values for the torus, for meshes of different average edge lengths. On the table, the error, maximum error, error after area correction, and maximum error after area correction are displayed.

Average Edge Length (h)	e	e_{max}	e_{ac}	$e_{ac_{max}}$
0.16	6.35%	2.04	6.31%	2.04
0.12	4.43%	1.31	3.88%	1.31
0.085	3.53%	1.17	3.69%	1.56
0.06	3.34%	1.41	3.34%	1.38
0.03	2.18%	1.08	2.17%	1.17
0.02	2.10%	1.86	2.19%	1.86

In tab. 4.6, the mean curvature error values are presented, with and without area correction, for meshes with average edge length $h_i = 0.16$ (393 nodes, 786 elements), $h_i = 0.12$ (698 nodes and 1396 elements), $h_i = 0.085$ (1363 nodes and 2726 elements), $h_i = 0.06$ (2697 nodes and 5394 elements), $h_i = 0.03$ (10549 nodes and 21098 elements), $h_i = 0.02$ (23540 nodes and 47080 elements). Mesh refinement offers improvement, but the area correction does not offer further enhancement. This is expected for meshes that do not have many obtuse triangles, but the values confirm that at worse, the area correction does not decrease mean curvature calculation accuracy.

When transitioning from an area of positive mean curvature to an area of negative mean curvature, some mean curvature values were very close to zero. Due to the division by a number close to zero, the numerical error when calculating the maximum relative curvature error was very high, masking the real maximum cur-

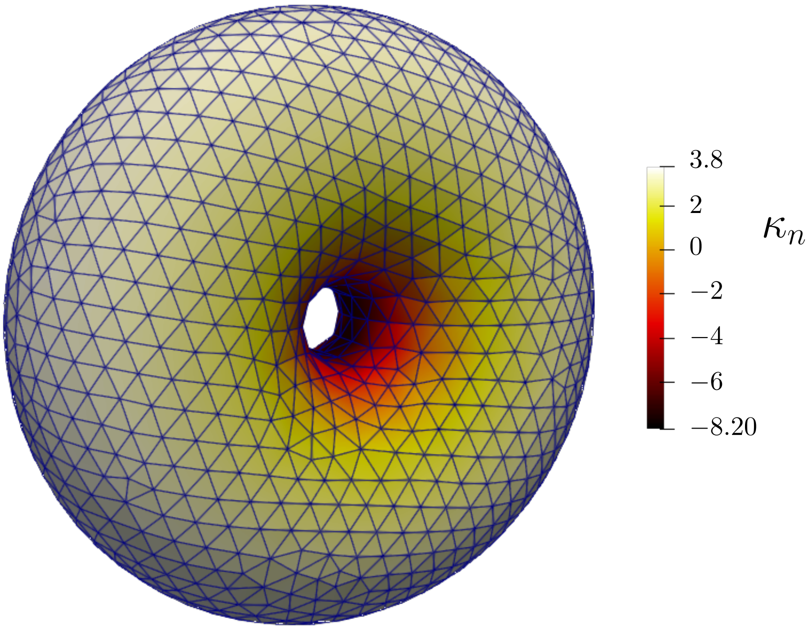


Figure 4.11: Torus with the major radius $R = 0.5$ and minor radius $r = 0.4$. The mesh average edge length is $h_i = 0.85$, and it has 1363 nodes and 2726 elements. In the scale, are represented the mean curvature values.

vature error. Therefore, the maximum error displayed in tab. 4.6 does not decrease with mesh refinement. These error results were calculated using equations 4.5.

4.3 Two-Phase Simulation

4.3.1 Static Droplet

For this test, a droplet of fluid is simulated in less dense, quiescent fluid, delimited by a spherical interface, in the absence of any external force or velocity field. No forces except the surface tension due to the interface are applied to the fluid. If the velocity fields are $\mathbf{v} = 0$ for the whole fluid domain, the Navier-Stokes equations are reduced to

$$\nabla p = \frac{1}{We}\mathbf{f} \tag{4.6}$$

The equation above is not the one solved by the numerical simulation, instead the simulation solves the complete Navier-Stokes equation, however, setting the boundary conditions to zero velocity and gravity forces to zero should make the terms not related to the pressure and surface tension vanish, and therefore the result should match the values obtained by equation 4.6. This is not, however, the case. Due to errors related to the mean curvature calculation, element discretization, and numerical errors, spurious velocities may appear and turn the simulation unstable. These spurious velocities are clearly undesired and should be minimized as possible. Lower intensity spurious velocities reflect a more accurate simulation.

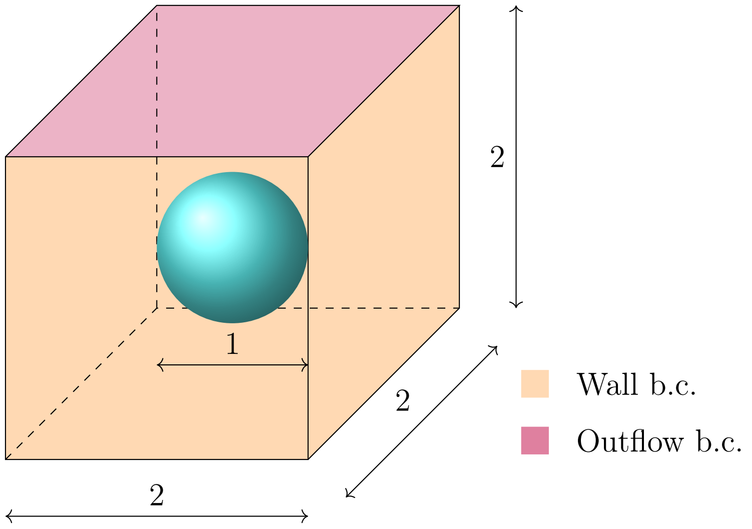


Figure 4.12: Finite element simulation parameters for the Static Droplet test case. A cubic domain is established, with no inflow conditions. At the walls, $u = v = w = 0$, except for the wall at the top, where outflow conditions were set, $p = 0$. The cubic domain's edge has length $l = 2$. The droplet has diameter $d = 1$.

For the background mesh, a cubic-shaped domain of $width = length = height = 2$ (non-dimensional units) was established, with an average edge length $h_f = 0.71$ (131188 nodes, 110464 elements). The interface mesh is a sphere of diameter $d = 1$. The fluid properties were set to $\mu = 1$ and $\rho = 1$ for the droplet, and $\mu = 0.01$ and $\rho = 0.001$ for the outside fluid. The Reynolds number and Weber number were set to $We = Re = 1$. The simulation was executed over 1200 time steps, $\Delta t = 0.005$, totaling 6 non-dimensional time units.

For the boundary conditions, at the top wall, outflow boundary conditions were used, with pressure set to $p = 0$. The other surfaces of the cube had prescribed velocity values set to $u = v = w = 0$. Several meshes were used for the interface, with average edge length $h_i = 0.16$ (159 nodes, 313), $h_i = 0.12$ (317 nodes, 630 elements), $h_i = 0.085$ (625 nodes, 1246 elements), $h_i = 0.06$ (1141 nodes, 2277 elements), $h_i = 0.03$ (4308 nodes, 8612 elements) and $h_i = 0.02$ (9500 nodes, 18996 elements). The intensity of the spurious velocities, along with the pressure calculation error can be observed in tab. 4.7.

According to the Young-Laplace equation given below in its non-dimensional form:

$$\Delta p = \frac{1}{We}\left(\frac{1}{R_1} + \frac{1}{R_2}\right) \tag{4.7}$$

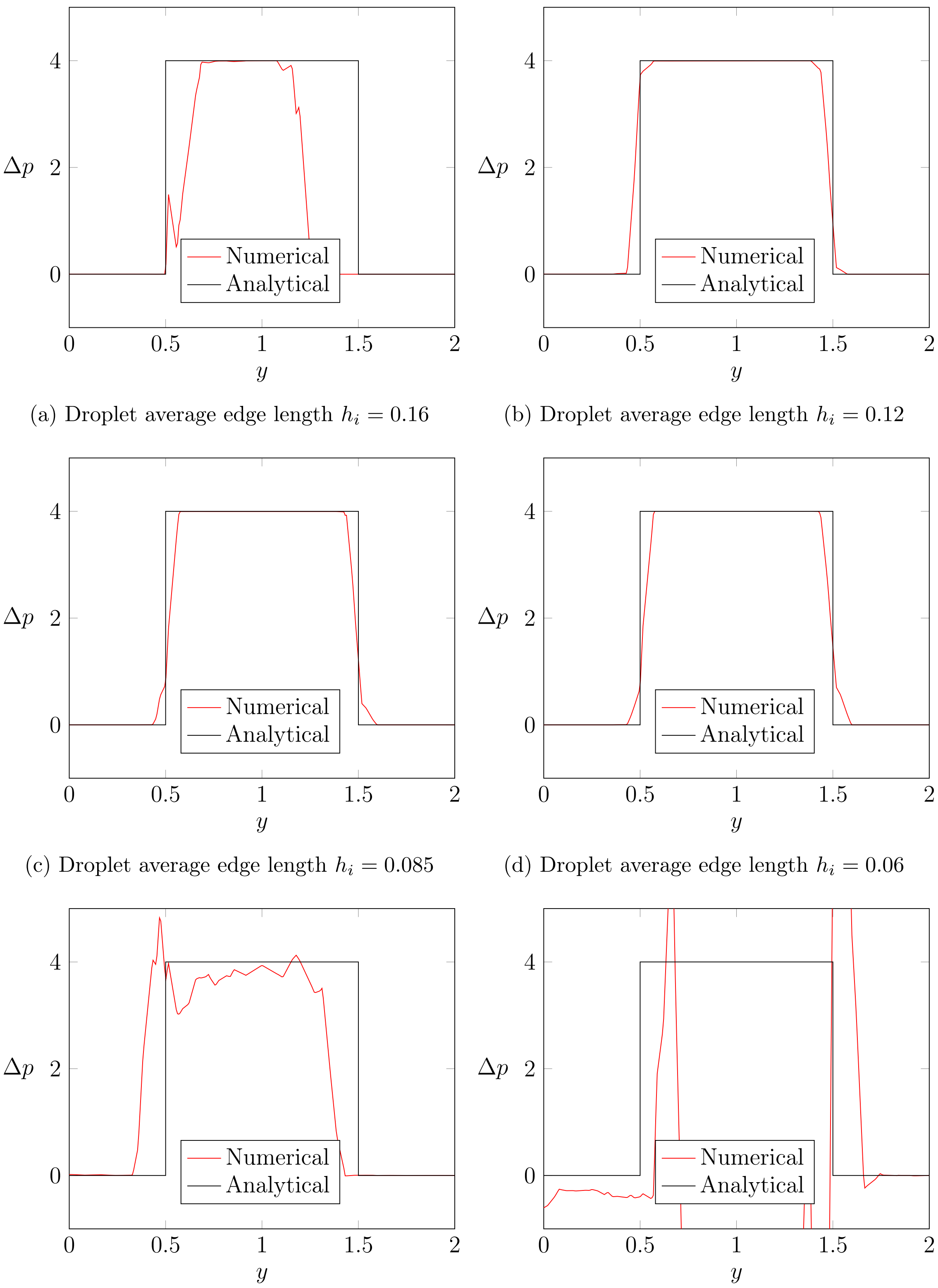
where R_1 and R_2 are the principal radii of curvature, for a droplet of radius $r = 0.5$, the pressure difference should be $\Delta p = 4$. The pressure across the simulation domain can be observed in fig. 4.13. The pressure difference is a close match to the analytical value.

Table 4.7: Spurious velocity intensity and pressure error for the static droplet test case. The test case was executed with the fluid mesh edge length $h_f = 0.071$ (131188 nodes, 110464 elements), on a domain of $length = width = height = 2$. The simulation appears to degrade when the interface mesh average edge length is smaller than the fluid mesh average edge length, optimal values are obtained with close average edge lengths.

Interface Edge (h)	v_{max}	Δp_{error}
0.16	6×10^{-1}	0.408%
0.12	1.9×10^{-2}	0.118%
0.085	1.2×10^{-2}	0.102%
0.06	4×10^{-3}	0.061%
0.03	4.2×10^{-1}	18.4%
0.02	366	1223.1%

After observing that the average edge length for the interface mesh of $h_i = 0.06$ gives the best results, the influence of boundary condition configurations was tested. Five different configurations were used, all of them using the interface mesh with $h_i = 0.06$, and the fluid mesh with $h_f = 0.071$. The five configurations were:

- Wall boundary conditions at all domain faces and outflow boundary conditions at domain corner points, totaling eight points with outflow conditions (corners).
- Wall boundary conditions at all domain faces and outflow boundary conditions at domain edges, totaling twelve edges with outflow conditions (edges).
- Wall boundary conditions at five domain faces and outflow boundary conditions at the remaining domain face (one face).
- Wall boundary conditions at four domain faces and outflow boundary conditions at the remaining two domain faces, opposing each other (two faces).



(a) Droplet average edge length $h_i = 0.16$

(b) Droplet average edge length $h_i = 0.12$

(c) Droplet average edge length $h_i = 0.085$

(d) Droplet average edge length $h_i = 0.06$

(e) Droplet average edge length $h_i = 0.03$

(f) Droplet average edge length $h_i = 0.02$

Figure 4.13: Static droplet test case pressure graph, from the cubic domain bottom face center $(1, 0, 1)$ to the domain's top face center $(1, 2, 1)$. Values obtained from interpolation of nodal values from the simulation with interface average edge length $h_i = 0.06$.

- Wall boundary conditions at three domain faces and outflow boundary conditions at the remaining three faces. Each face that is assigned a wall boundary condition opposes a face with an outflow boundary condition (three faces).

The spurious velocity values for each of the five boundary condition configurations can be observed in tab. 4.8. The configuration which offered the best (lowest) spurious velocity values was the third one, represented in fig. 4.12

Table 4.8: Spurious velocity intensity for the static droplet test case with different boundary condition configurations. The test case was executed with the fluid mesh average edge length of $h_f = 0.071$, average interface mesh length of $h_i = 0.06$, and on a domain of $length = width = height = 2$, droplet radius $r = 0.5$.

Boundary Condition Configuration	v_{max}
Outflow at corners	1×10^{-2}
Outflow at edges	1×10^{-2}
Outflow in one face	4×10^{-3}
Outflow in two faces	2.5×10^{-2}
Outflow in three faces	3.3×10^{-2}

Based on tab. 4.8, the best boundary condition configuration is the one adopted initially, a single face with outflow boundary conditions assigned, and the remaining faces assigned wall boundary conditions.

In addition to the tests executed with varying interface mesh refinement levels and different boundary condition configurations, a test altering both the fluid mesh refinement and the interface refinement was also executed. Seven simulations were carried out, with fluid mesh refinement, interface mesh refinement and time step sizes detailed in Tab. 4.9. The results for maximum and average spurious velocities at the last iteration are displayed in fig. 4.14

Table 4.9: Spurious velocity intensity for the static droplet test case with various fluid mesh refinement levels. All simulations were executed on the same cubic domain of edge size 2, with a number of time steps totalling 5 non-dimensional time units. Interface mesh and time steps were chosen based on the fluid mesh refinement levels.

No. of fluid nodes	Fluid Mesh (h_f)	Interface Edge (h_i)	Time Step	v_{max}	$v_{average}$
15389	0.145	0.15	0.0145	1.40×10^{-2}	3.36×10^{-3}
32247	0.115	0.12	0.0115	6.66×10^{-3}	1.85×10^{-3}
66641	0.090	0.09	0.0090	4.30×10^{-3}	1.37×10^{-3}
131747	0.069	0.07	0.0069	6.57×10^{-3}	2.09×10^{-3}
269847	0.055	0.06	0.0055	2.50×10^{-3}	8.68×10^{-4}
546853	0.043	0.045	0.0043	1.81×10^{-3}	3.12×10^{-4}
1132910	0.0335	0.035	0.00335	1.64×10^{-3}	4.93×10^{-4}

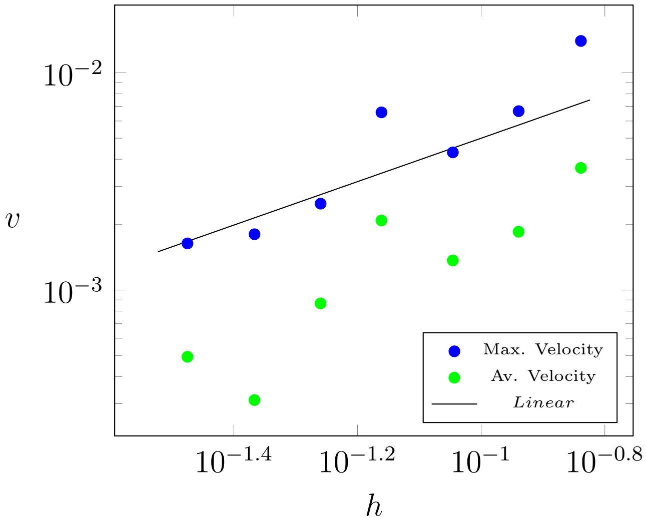


Figure 4.14: Maximum and average velocities are presented for various fluid mesh refinement levels in comparison with a linear function ax. The simulation details for these tests are detailed in Tab. 4.9.

4.3.2 Oscillating Droplet

In this test case, a droplet with an ellipsoidal shape suspended in a less dense fluid is simulated, with no gravity force acting upon the fluids. The droplet starts with an ellipsoidal shape, and should oscillate around its equilibrium radius, being damped by viscosity. Over enough time, the droplet should approach a sphere with radius equal to the equilibrium radius.

The goal of this test is to evaluate the behavior of the surface tension force comparing the oscillation frequency of the simulated droplet against the known exact oscillation frequency. If the oscillation frequency matches the exact frequency closely, it indicates that simulation's surface tension force is represented accurately.

The exact oscillation frequency is known, it is detailed in [73] and is given by:

$$d = d_o + cos(\omega t) a_o e^{\dfrac{-5\mu_{in} t}{r \rho_{in}}} \tag{4.8}$$

$$\omega = \sqrt{\frac{24}{(3\rho_{in} + 2\rho_{out}) r^3 We}} \tag{4.9}$$

d is the diameter, d_o is the equilibrium diameter, a_o is the initial perturbation, t is time, r is the equilibrium radius and ω is the oscillation frequency.

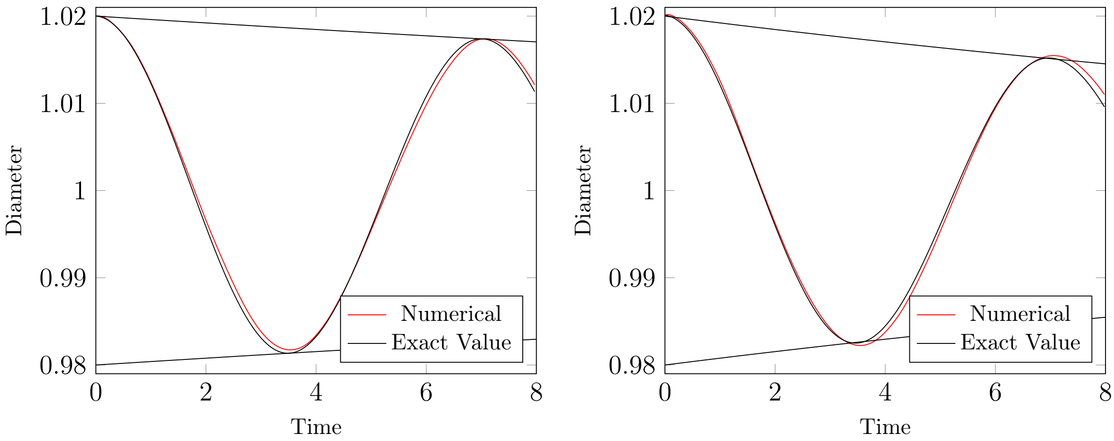
The exact solution assumes the outer fluid's influence is negligible and second order effects are small enough to be ignored. To simulate those conditions numerically, the outer fluid properties were set to very small values, as to have the outer fluid influence the results as little as possible. The ellipsoidal droplet's diameters were selected to represent a small perturbation while maintaining the same volume as an spherical droplet of diameter $d = 1$, which is its equilibrium diameter. The diameters selected are $a = 1.02$, $b = c = 0.985246$.

The test setup is as follows, the domain is the same as the static droplet test case, a cube of edge size 2, with the droplet positioned at its center. No slip boundary condition was set on all the cube's walls, and homogeneous Dirichlet pressure boundary condition was set at the cube's corners. Inner fluid properties are set to $\rho = 1$ and $\mu = 1$ and outer fluid properties set to $\rho = 0.0001$ and $\mu = 0.0001$.

Two simulations were executed, both of them using the same fluid mesh. This fluid mesh has non uniform element sizes to save on simulation time. An spherical region of diameter $d = 1.04$ located at the fluid mesh center has an average edge length of $h_f = 0.02$, transitioning smoothly to the rest of the domain with an average edge length of $h_f = 0.2$. This results on a mesh with 594381 nodes and 511318 tetrahedra.

The first simulation had the Reynolds number set at $Re = 1000$ and Weber number at $We = 80$, an interface mesh with $h_i = 0.27$, composed of 5443 nodes and 10882 triangles. The second simulation had a Reynolds number set at $Re = 500$, the same Weber number $We = 80$ and the interface mesh with $h_i = 0.28$, composed of 5485 nodes and 10966 triangles. Both simulations were executed over 8 non dimensional time units and had the interface thickness set to $\epsilon = 0.6h_f = 0.0218259$. The first simulation used a time step of $\Delta t = 0.032$ over 250 time steps and the second with a time step of $\Delta t = 0.02$ over 400 time steps. No remeshing procedure was executed for both simulations, but to increase simulation stability, the mesh filter detailed by [74] was utilized.

The obtained results can be observed in fig. 4.15. The result's amplitude presented itself very dependent upon the time step size. Increasing the time step can increase the amplitude of the results, while decreasing the time step diminished it for most tests. This test is also sensitive to the interface thickness choice, affecting the oscillation frequency. Generally, a smaller interface thickness produces better results, but the optimal interface thickness changes with changes in other parameters, such as the Reynolds number. Careful selection of parameters was necessary to obtain the results presented in fig. 4.15. The average error was calculated for both simulations using the formula given below



(a) Oscillating Droplet with $Re = 1000$. (b) Oscillating Droplet with $Re = 500$.

Figure 4.15: Oscillating Droplet results, compared to exact data. In a) the test was executed with $Re = 1000$ over 250 time steps and in b) with $Re = 500$ over 533 time steps, both cases simulating 8 seconds total, with the Weber number set to $We = 80$. The fluid mesh has an average length $h_f = 0.02$ at the most refined region, with 594381 nodes and 511318 tetrahedra. The test in a) presented an error of $e_{av} = 5.94\%$, while the test in b) has an error of $e_{av} = 5.63\%$.

$$e_{av} = \frac{\sum \left| \dfrac{(d_n - 1) - (d_e - 1)}{(d_e - 1)} \right|}{n} \tag{4.10}$$

where d_n represents the diameter of the droplet in the z direction obtained from the simulation at the current time step, d_e is the exact value of the droplet diameter for the current time step, and n is the total number of time steps. Using the described formula, an average error of $e_{av} = 5.94\%$ was found for the first simulation, presented in fig. 4.15a, and $e_{av} = 5.63\%$ for the second simulation presented in fig. 4.15b.

4.3.3 Rising of an air bubble in a sugar-water solution

This test consists in simulating an spherical bubble inside a quiescent fluid, subjected to gravity force. The bubble's fluid is less dense than the outside fluid, and as the bubble rises, it changes shape. The shape the bubble takes and its terminal velocity depend upon the flow properties, expressed by its Archimedes and Eötvös numbers. Both the bubble's shape and terminal velocity can be compared to experimental data in order to assess the two-phase simulation accuracy. The air bubble in an sugar-water solution experiment described in [75] was selected as reference to compare the simulated bubble shape and terminal velocity.

The following procedure was adopted to find the non-dimensional velocity used as reference. For the selected flow, [75] reports a Reynolds number of $Re = 7.16$, for a bubble of volume $V = 9.3 \times 10^{-6} m^3$ at terminal velocity. Replacing the sugar water solution density $\rho = 1350 kg/m^3$, viscosity $\mu = 1.28 kg/ms$, and the bubble's diameter of $D = 0.00261m$ obtained from its volume into the Reynolds number equation, a dimensional velocity $u = 0.26m/s$ is obtained. For a gravity driven flow, the non dimensional velocity is given by:

$$\mathbf{v}^* = \frac{\mathbf{v}}{\sqrt{g_o D}} \tag{4.11}$$

where $\mathbf{v}*$ is the non-dimensional velocity, $\mathbf{v}$ is the dimensional velocity, D is the bubble's diameter and g_o is the reference gravity acceleration. By replacing the these values into the described formula, a value for the non-dimensional velocity $\mathbf{v}^* = 0.514$ was found.

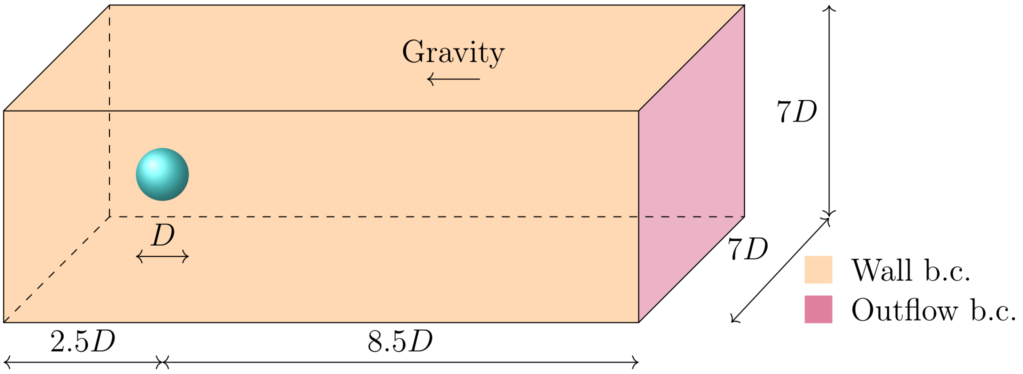


Figure 4.16: Finite element simulation domain parameters for the Rising Bubble test case. A prismatic domain is established, with no inflow condition. At the walls, $u = v = w = 0$, except for the wall to the right, where outflow condition was set, $p = 0$.

The numerical simulation fluid domain can be observed in fig. 4.16. It is a prismatic channel with a square bottom and the following dimensions, $width = height = 7D$ and $length = 11D$. An spherical bubble of diameter $d = 1$ has its center positioned at $x_o = (2.5, 3.5, 3.5)$. The outer fluid properties are $\rho = \mu = 1$ while the inner fluid properties are $\rho = 0.000892$ and $\mu = 0.0000142$. The Archimedes number is $N = 194.88$ and the Eötvös number is $Eo = 115.662$. It should be noted that the property ratios are higher than 10^3.

The numerical parameters adopted were a time step of $\Delta t = 0.015$, 400 time steps executed simulating 6 non-dimensional time units. Several interface thickness values were adopted, $\epsilon = 1.0h_f$, $\epsilon = 1.6h_f$, $\epsilon = 2.2h_f$ and $\epsilon = 2.8h_f$. The fluid mesh utilized is non uniform, with its more refined region possessing an average edge length of $h_f = 0.06$, totalling 1012381 nodes and 871124 tetrahedra. The interface mesh has an average edge length of $h_i = 0.07$, and started with 829 nodes and 1654 triangles.

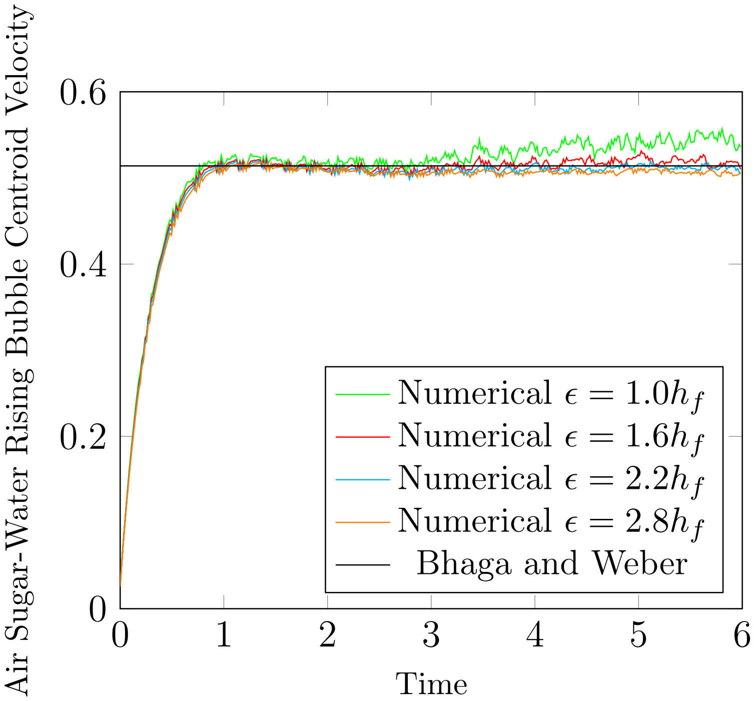
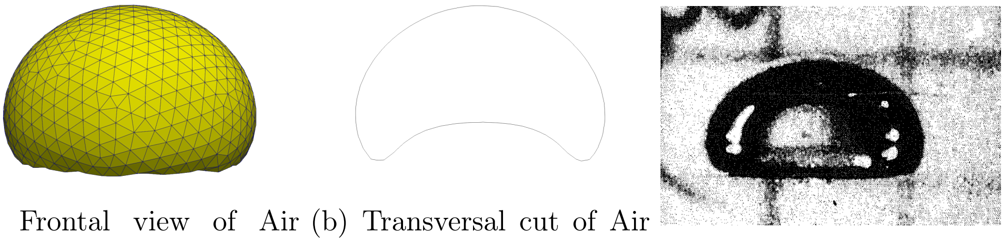


Figure 4.17: Finite element simulation results for the air bubble inserted in a sugar-water solution, compared to experimental results. The fluid property ratios are $\rho_{in}/\rho_{out} = 0.000892$, $\mu_{in}/\mu_{out} = 0.0000142$, and the relevant non-dimensional numbers are $N = 194.88$ and $Eo = 115.662$. The fluid domain is comprised of 1012381 nodes and 871124 tetrahedra, and the interface mesh 829 nodes and 1654 triangles. The simulation was executed over 400 time steps of $\Delta t = 0.015$.

The bubble's terminal velocity obtained from the air sugar-water numerical simulation is plotted against time in fig. 4.17, along with the terminal velocity measured by [75]. The shape produced by the simulation at the last iteration can be compared to the actual bubble in fig. 4.18. Good agreement is observable both in terminal velocity and bubble shape.

4.3.4 Falling Droplet in Inert Media

The goal of this test is to simulate a droplet of fluid in free fall, and compare its position and velocity to exact values obtained by the equations of motion. To achieve this, a droplet of dense fluid is placed in a very large domain, inserted in less dense and less viscous fluid, to minimize the drag effects acting upon the droplet.



(a) Frontal view of Air Sugar-Water bubble at *iter* = 400.

(b) Transversal cut of Air Sugar-Water bubble at *iter* = 400.

(c) Picture of Air Sugar-Water bubble

Figure 4.18: In (a), the simulated bubble shape can be observed. In (b), a contour of the middle of the bubble is represented, where the dimple at the bottom can be observed. In (c), a picture of the actual bubble is displayed.

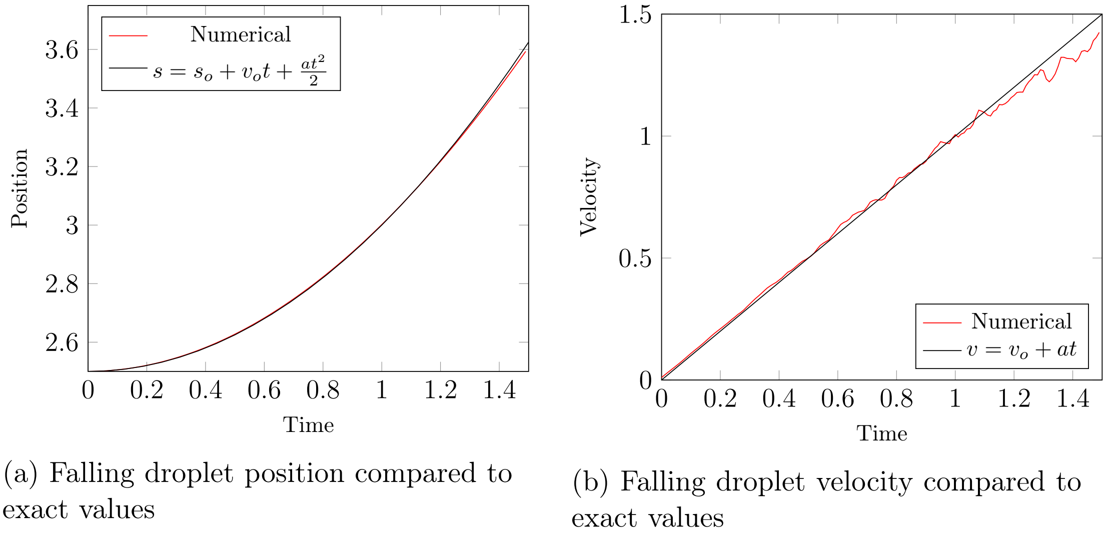
The simulation parameters are as follows, a droplet of diameter $d = 1$ with its center positioned at $x_o = (2.5, 3.5, 3.5)$, inside a domain of dimensions $width = height = 7D$ and $length = 11D$, the same set up as in fig. 4.16, but with gravity pointing in the positive x direction. The inner fluid properties are $\mu_{in} = \rho_{in} = 1$ and the outer fluid properties are $\mu_{out} = \rho_{out} = 0.0001$. Archimedes and Eötvös numbers are $N = 1000000$ and $Eo = 1$, and a time step of $dt = 0.01$ was used over 150 time steps, simulating 1.5 non-dimensional time units. A non-uniform mesh was used, with 1012381 nodes, and a refinement of $h_f = 0.06$ at its most refined section (the center, along the expected droplet path). The uniform interface mesh has 625 nodes, with a constant refinement level of $h_i = 0.08$. The selected interface thickness was $\epsilon = 1.4h_f$. Non-dimensional gravity acceleration of $g = 1m/s^2$ is utilized.

The results of the droplet's position and velocity can be observed in fig. 4.19. The results present an exact match up until $t = 1$. At the end of the simulation, close to $t = 1.5$, some minor discrepancy can be observed, which can be attributed to increasing drag forces acting upon the droplet.

4.3.5 Rising of an air Taylor bubble

In this section we present the results obtained by simulating an elongated Taylor bubble in a circular channel. In this type of flow the channel's walls influence the bubble's shape, and the liquid film thickness, the distance between the channel's walls to the bubble, is an important parameter. Therefore the results of this simulation will be compared against the terminal velocity for a Taylor bubble presented by [76], and a correlation for liquid film thickness obtained from lubrication theory shown in [77].

The domain for the numerical simulation is presented in fig. 4.20. It is a cylinder of diameter $d = 1$, and length $l = 11$. Inside this domain, an approximately bullet-shaped bubble is placed with its end positioned at 2 diameters from the base of the



(a) Falling droplet position compared to exact values

(b) Falling droplet velocity compared to exact values

Figure 4.19: In (a), the position of the falling droplet is compared against the exact values. In (b), the droplet's velocity is displayed against the precise values derived from the equations of motion. Gravity points in the positive x direction. The noise observed in (b) can be attributed to numerical errors arising from the computational methods used in velocity calculations.

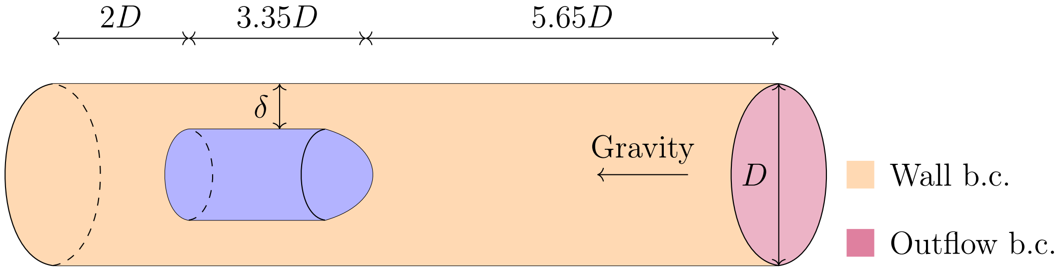


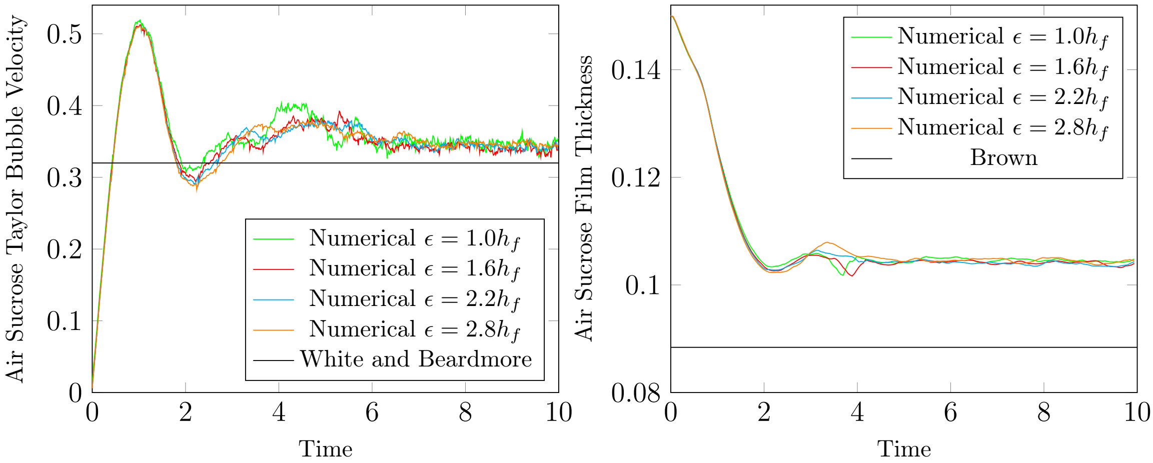
Figure 4.20: Finite element simulation domain parameters for the Taylor bubble test case. A cylindrical domain is established, with no inflow condition. At the cylinder's base and walls, $u = v = w = 0$. At the top outflow condition was set, $p = 0$.

cylinder, centered along the cylinder's axis. The same setup is used for two different fluids, a sucrose solution and an ethylene glycol solution.

In the sucrose solution simulation the outer fluid properties are $\rho = \mu = 1$, while the inner fluid properties are $\rho = 0.00101024$ and $\mu = 0.00323$. The Archimedes number is $N = 800000$ and the Eötvös number is $Eo = 40$.

In the ethylene glycol solution simulation the outer fluid properties are $\rho = \mu = 1$, while the inner fluid properties are $\rho = 0.0010638$ and $\mu = 0.0009171$. The Archimedes number is $N = 107439$ and the Eötvös number is $Eo = 53$.

For both simulations, a time step of $\Delta t = 0.008$ was adopted, 1250 time steps were executed simulating 10 non-dimensional time units. Four interface thickness were tested, $\epsilon = 1.0h_f$, $\epsilon = 1.6h_f$, $\epsilon = 2.2h_f$ and $\epsilon = 2.8h_f$. The fluid mesh utilized has the same refinement over the whole domain, with an average edge length of



(a) Air sucrose Taylor bubble center of mass non dimensional velocity

(b) Air sucrose Taylor bubble film thickness

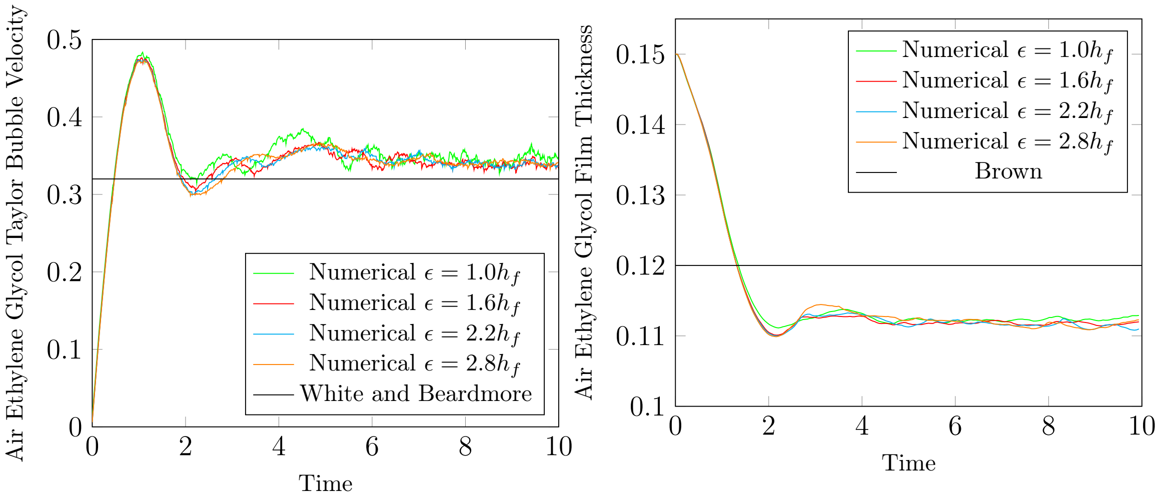
Figure 4.21: Finite element simulation results for the air bubble inserted in a sucrose solution, compared to experimental results. The fluid property ratios are $\rho_{in}/\rho_{out} = 0.00101024$, $\mu_{in}/\mu_{out} = 0.00323$, and the relevant non-dimensional numbers are $N = 800000$ and $Eo = 40$. The fluid domain is comprised of 709579 nodes and 600375 tetrahedra, and the interface mesh 4070 nodes and 8136 triangles. The simulation was executed over 1250 time steps of $\Delta t = 0.008$.

$h_f = 0.04$, totalling 709579 nodes and 600375 tetrahedra. The interface mesh has an average edge length of $h_i = 0.04$, starting with 4070 nodes and 8136 triangles.

In fig. 4.21 and fig. 4.22, the results for the sucrose solution and ethylene glycol solutions can be observed, respectively, along with the values for film thickness for each test. The results are plotted against the bubbles' terminal velocity obtained experimentally in [76], and in the same figures, the liquid film thickness is plotted against the values obtained by the equation described by [77] presented below:

$$\delta^3 = 1.5Fr\frac{Mo^{0.25}}{Eo^{0.75}}(0.5 - \delta) \tag{4.12}$$

where Fr, Mo and Eo are the non dimensional numbers presented previously. The equation was solved numerically using Newton's method. Both fluids simulated show results with reasonable agreement to experimental data, both in terminal velocity and liquid film thickness. It is apparent that the interface thickness is not a major influence in these results, where the gravity force is more significant than the surface tension force. The reported velocity and film thickness values at the start of the simulation show a major disagreement to exact values due to the bubble's shape



(a) Air ethylene glycol Taylor bubble center of mass non dimensional velocity

(b) Air ethylene glycol Taylor bubble film thickness

Figure 4.22: Finite element simulation results for the air bubble inserted in ethylene glycol, compared to experimental results. The fluid property ratios are $\rho_{in}/\rho_{out} = 0.0010638$, $\mu_{in}/\mu_{out} = 0.0009171$, and the relevant non-dimensional numbers are $N = 107439.7$ and $Eo = 53$. The fluid domain is comprised of 709579 nodes and 600375 tetrahedra, and the interface mesh 4070 nodes and 8136 triangles. The simulation was executed over 1250 time steps of $\Delta t = 0.008$.

adjustment. A significant modification of the bubble shape happens before it turns stable, and that is reflected in the presented results.

4.3.6 Taylor Bubble in Complex Geometry

Using the different geometry proposed in the previous section, several Taylor bubble simulations in different channel cross-sections were executed. A square, triangular, hexagonal and ellipsoidal cross-sections were selected, each with the same hydraulic diameter as a circular cross-section of diameter $d = 1$. The square cross-section features an edge length of $l = 1$, while the hexagonal cross-section has a side length of $l = 0.577$, the triangular cross-section, which is an equilateral triangle, possesses a side length of $l = 1.73$, and the ellipsoidal cross-section has a minor diameter of $a = 0.842$ and a major diameter of $b = 1.262$.

The fluids selected were air and ethylene glycol, with the same properties and simulation parameters as described previously section, and air and tellus oil, an hydraulic fluid, which possesses the following simulation parameters, the outer fluid properties are $\rho = \mu = 1$, while the inner fluid properties are $\rho = 0.00137037$ and $\mu = 0.000350962$. The Archimedes number is $N = 11094$ and the Eötvös number is $Eo = 70$. The time step selected was $\Delta t = 0.01$ over 1000 time steps and the interface thickness selected was $\epsilon = 1.2h_f$.

The setup is the same as represented in fig. 4.20, with the same bubble shape and length and same channel dimensions, the only difference is the channel cross-section, which follows the geometries in fig. 4.6.

Figure 4.23 presents the bubbles' velocities for each cross-section of the ethylene glycol simulation. The simulation for the triangular cross-section did not complete successfully. In the triangular simulation, the bubble tends to undergo breakup during the shape adjustment phase, a phenomenon not accounted for in the simulation.

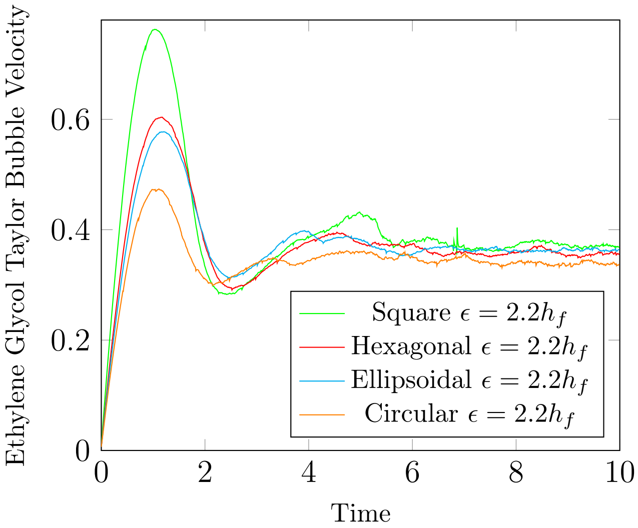


Figure 4.23: Terminal velocity comparison for different cross-sections with the same hydraulic diameter. Ethylene Glycol is the chosen fluid.

Figure 4.24 presents the bubbles' velocities for each of the cross-sections presented in fig.4.6, for the tellus oil fluid. The more viscous nature of the tellus oil restricted bubble deformation during the initial shape adjustment phase, and all simulations finished successfully.

Geometric changes in the air bubbles can be observed in simulations with both fluids, but the less viscous ethylene glycol presents more dramatic changes, which can be observed in fig. 4.25. The bubbles take a very smoothed shape of the channel they are in. The geometry changes are barely noticeable on the hexagonal cross-section, but they are significant on the square and ellipsoidal channel cross-sections. The bubble skirts are more accentuated in regions close to the corners of the fluid domain, an effect very noticeable on the square and elliptic channels.

4.3.7 Gas Bubble Interaction in a Shear Flow

In this section, the results for multiple bubbles rising in fluid with varying Reynolds and Weber numbers are presented. All tests executed and presented in this section employed the fluid properties of carbon dioxide (CO_2) for the gas phase and crude-oil for the liquid phase. Carbon dioxide's specific mass is $\rho_{in} = 0.1$ kg/m^3, and

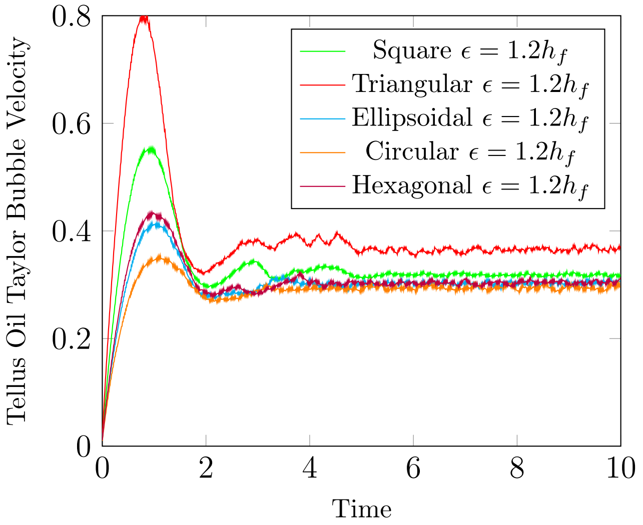


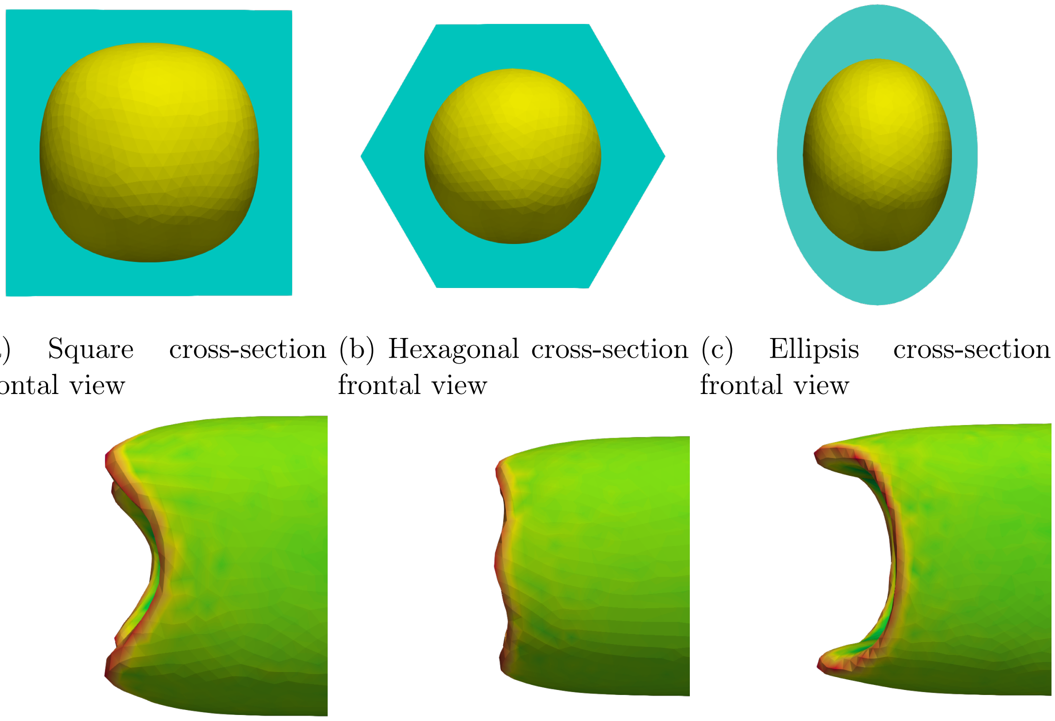
Figure 4.24: Terminal velocity comparison for different cross-sections with the same hydraulic diameter. Tellus oil is the chosen fluid.

its viscosity is $\mu_{in} = 0.1$ kg/ms. The crude-oil properties can vary over a large range, depending on the oil components. It can be a lighter-than-water, low-viscosity fluid or a heavy, almost-solid substance. In this work, the following average properties were considered: specific mass of $\rho_{out} = 910$ kg/m^3 and viscosity of $\mu_{out} = 12.85$ kg/ms. The surface tension coefficient also varies with the oil type, but an average value of $\sigma = 17$ N/m is considered.

The simulation setup can be observed in fig. 4.26. A vertical circular channel is represented, with a non-dimensional diameter $d = 1$ and length $l = 8$. At one end, an inflow with a uniform non-dimensional velocity $v = 1$ is set, and at the other end, an outflow where pressure $p = 0$ is positioned. The walls along the channel have no slip boundary conditions. Two bubbles are positioned inside the channel. The bubble at the center, labeled the free bubble, has its center positioned at $p_1 = (1, 0, 0)$, while the bubble at the side, labeled the shear bubble, has its center positioned at $p_2 = (1.3, 0, -0.31)$. Both bubbles are initially spherical in shape and have a diameter of $D_1 = D_2 = 0.3$. For reference, the center of the channel's bottom is positioned at $O = (0, 0, 0)$.

In relation to the initial positioning of the bubbles, our objective was to examine the influence of a bubble's wake on nearby bubbles within a circular channel. The positions of the bubbles within the channel were deliberately selected so that they have different velocities, enabling one bubble to overtake the other. Additionally, the diameters were determined to ensure that, if the bubbles maintain a straight trajectory along the x-axis, they would closely approach each other at the moment of passing.

By inputting the fluid properties specified in the previous paragraph into the



(a) Square cross-section frontal view (b) Hexagonal cross-section frontal view (c) Ellipsis cross-section frontal view

(d) Square cross-section lateral view (e) Hexagonal cross-section lateral view (f) Ellipsis cross-section lateral view

Figure 4.25: In (a), (b) and (c), the frontal view for the bubbles in the square, hexagonal and ellipsoidal cross-sections are displayed. It's observable that the bubbles take an approximate, smooth shape of the channel they are in. (d), (e) and (f) display the bubbles' lateral view. From these views, one can notice the skirt formation, that matches the positions of the channel's corners.

Morton number equation, three pairs of Reynolds and Weber numbers were selected to be simulated. The first one is $We = 10$ and $Re = 64$, the second one is $We = 20$ and $Re = 107$, and the third one is $We = 40$ and $Re = 180$. All three cases used $Fr = 1$. Simulations with those parameters were executed with and without gravity forces, totaling six scenarios. All simulations were conducted over 300 time steps of $dt = 0.01$, resulting in $t = 3$ non-dimensional time units. Using the properties presented previously, the following non-dimensional fluid properties were inputted, the outer fluid properties are $\mu_{out} = \rho_{out} = 1$, and the inner fluid properties are $\mu_{in} = 0.00155642$ and $\rho_{in} = 0.00078022$. The selected interface thickness was $\epsilon = 1h_f$. The fluid domain has 1218138 nodes, with an average edge length of $h_f = 0.3$, and the interface mesh for both bubbles had 760 nodes, with $h_i = 0.3$.

Figure 4.27 presents the results of the simulation executed with Weber and Reynolds $We = 10$ and $Re = 64$ at $t = 0.2$, $t = 0.9$, $t = 1.4$, and $t = 1.9$. For the given parameters, there is very little difference between the results obtained with and without gravity forces. In the simulation executed with gravity, the bubbles travel further ahead in the channel, but the shape and position relative to one

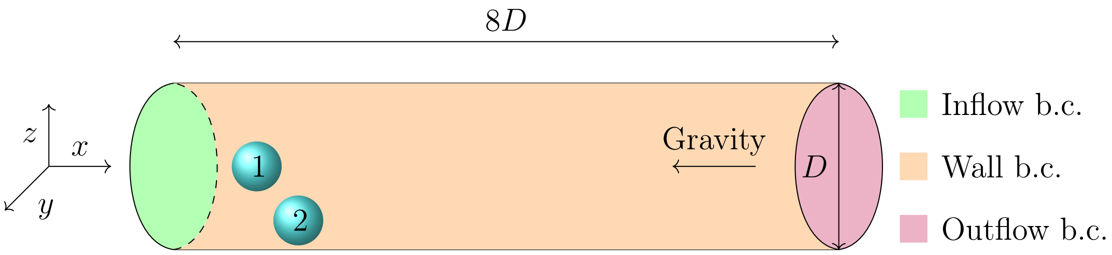


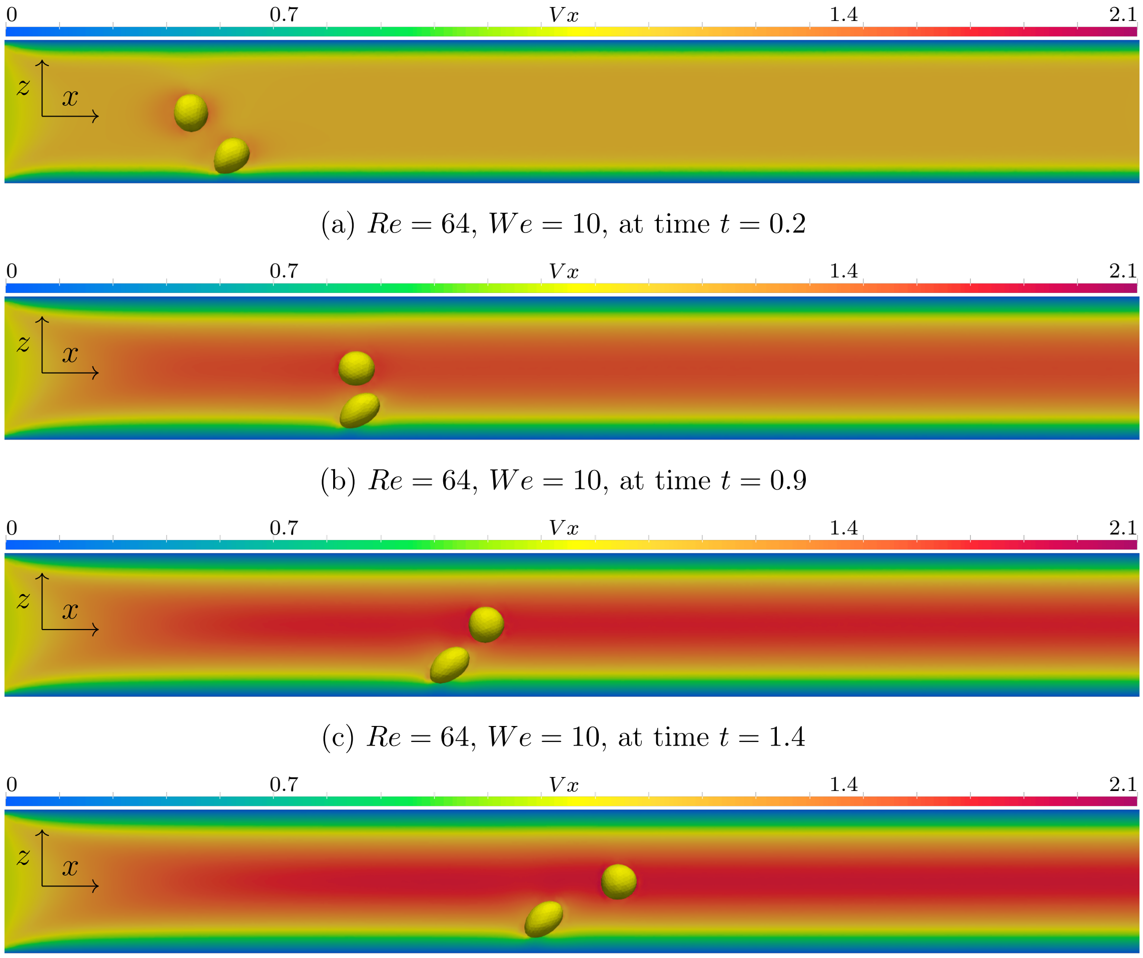
Figure 4.26: Simulation setup for the composed geometry, the representation of the continuous and dispersed phases, and the boundary conditions. The channel is oriented vertically from left to right but depicted in a rotated configuration for convenience to fit within the paper sheet. As a result, the gravity field acts in the opposite direction, from right to left. At the inflow, a prescribed velocity boundary condition is assigned, $v_x = 1, v_y = 0, v_z = 0$. At the outflow, an homogeneous Dirichlet pressure boundary condition is assigned, $p = 0$. At the cylinder walls, the no-slip boundary condition is set, $v_x = v_y = v_z = 0$.

another are very similar, as can be seen in fig. 4.28. The bubble near the wall undergoes some deformation due to the shear forces caused by the velocity gradient. There is very little influence on the free-bubble trajectory due to the shear bubble. After the free bubble passes the shear bubble, it is carried by the flow normally.

Figure 4.29 presents the results of the simulation executed with the Reynolds number $Re = 107$ and the Weber number $We = 20$, with no gravity forces, while fig. 4.30 shows the results with the same parameters but with gravity forces active. The times presented are $t = 0.2$, $t = 1.2$, $t = 1.8$, and $t = 2.4$. The results for the bubble simulation with no gravity forces are very similar to those of the previous simulations with $Re = 64$ and $We = 10$.

However, when gravity forces are active, there is some variation. The shear bubble suffers significantly more deformation and approaches the free bubble at the moment the free bubble passes it. The free bubble interacts with the shear bubble, changing shape as it moves along the flow, and the shear bubble appears to be trapped in the free bubble's wake, increasing its velocity compared to the simulation with no gravity forces active and approaching the center of the channel. This effect can be observed in fig. 4.31, along with the velocity fields that originate it.

Figure 4.32 presents the results of the simulation executed with the Reynolds number $Re = 180$ and the Weber number $We = 40$, with no gravity forces, while fig. 4.33 shows the results with the same parameters but with gravity forces active. The times presented are $t = 0.2$, $t = 1.2$, $t = 1.8$, and $t = 2.4$. The simulation with no gravity presents similar results to those of other simulations without gravity, with slightly more deformation for both bubbles due to lower viscosity and lower surface tension forces. The simulation executed with gravity forces turned on offers some interesting results. The shear bubble undergoes the most deformation, being



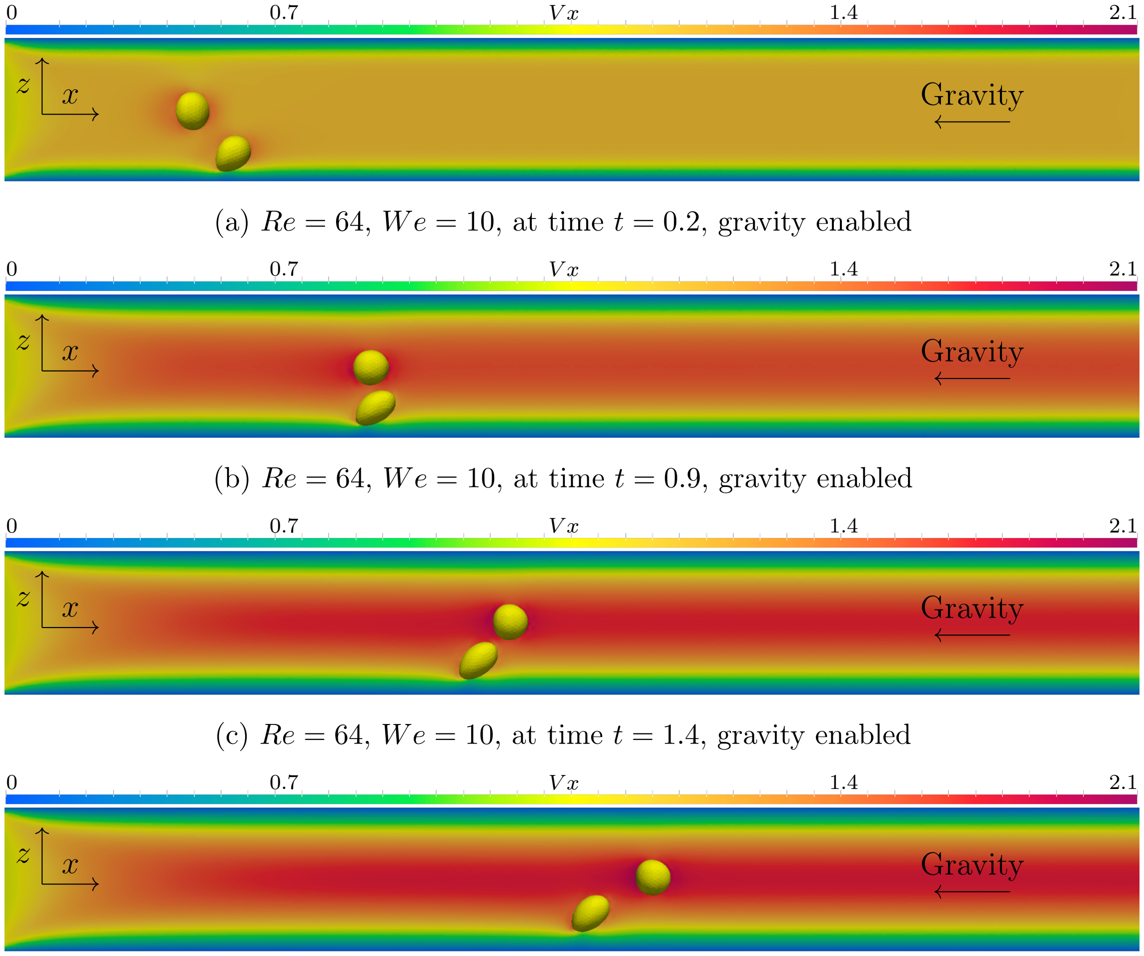
(a) $Re = 64$, $We = 10$, at time $t = 0.2$

(b) $Re = 64$, $We = 10$, at time $t = 0.9$

(c) $Re = 64$, $We = 10$, at time $t = 1.4$

(d) $Re = 64$, $We = 10$, at time $t = 1.9$

Figure 4.27: In a), b), c) and d), the simulation with parameters $Re = 64$ and $We = 10$ is displayed at times $t = 0.2$, $t = 0.9$, $t = 1.4$, and $t = 1.9$. V_x represents the axial velocity component. Gravity forces were disabled, and the simulation was executed over 300 time steps of $dt = 0.01$.



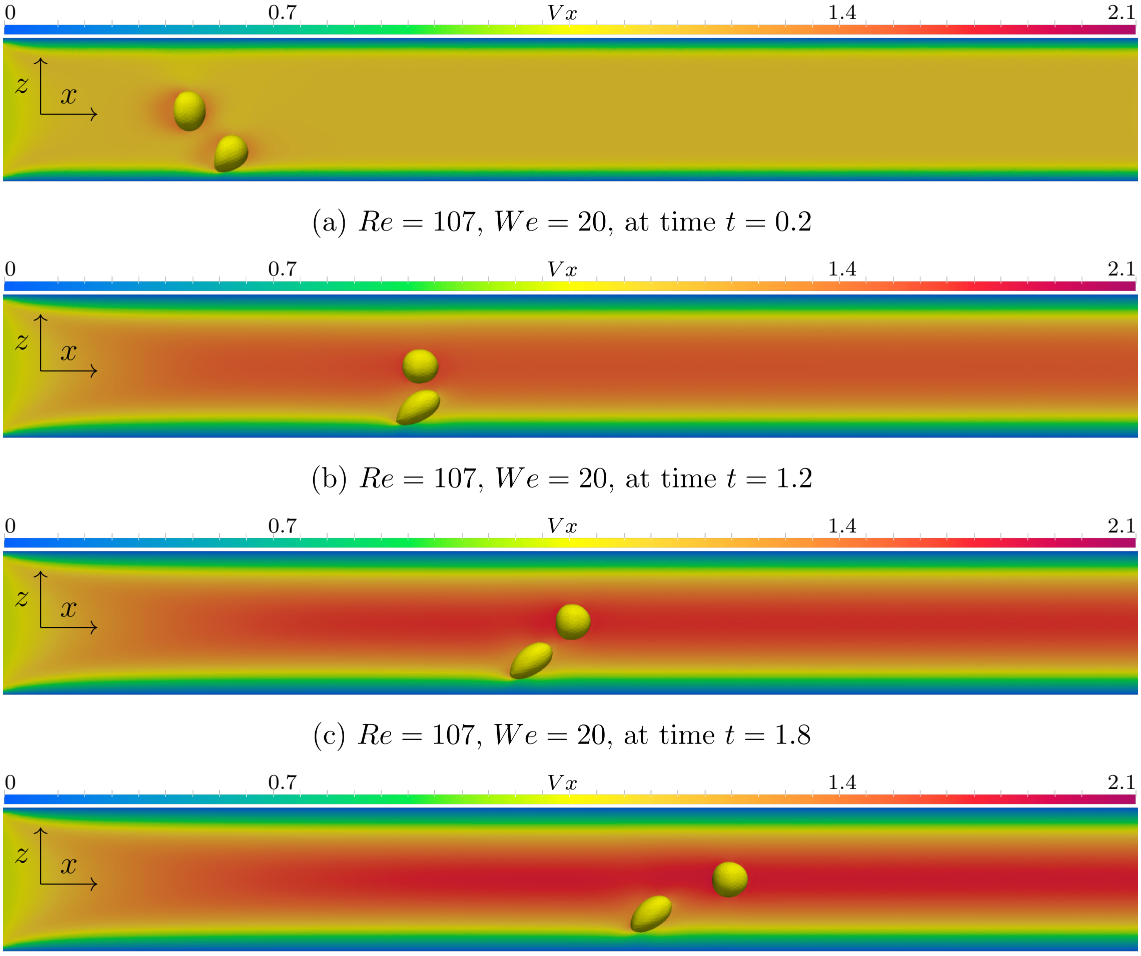
(a) $Re = 64$, $We = 10$, at time $t = 0.2$, gravity enabled

(b) $Re = 64$, $We = 10$, at time $t = 0.9$, gravity enabled

(c) $Re = 64$, $We = 10$, at time $t = 1.4$, gravity enabled

(d) $Re = 64$, $We = 10$, at time $t = 1.9$, gravity enabled

Figure 4.28: In a), b), c) and d), the simulation with parameters $Re = 64$ and $We = 10$ is displayed at times $t = 0.2$, $t = 0.9$, $t = 1.4$, and $t = 1.9$. V_x represents the axial velocity component. Gravity forces, pointing to the left, were enabled, and the simulation was executed over 300 time steps of $dt = 0.01$.



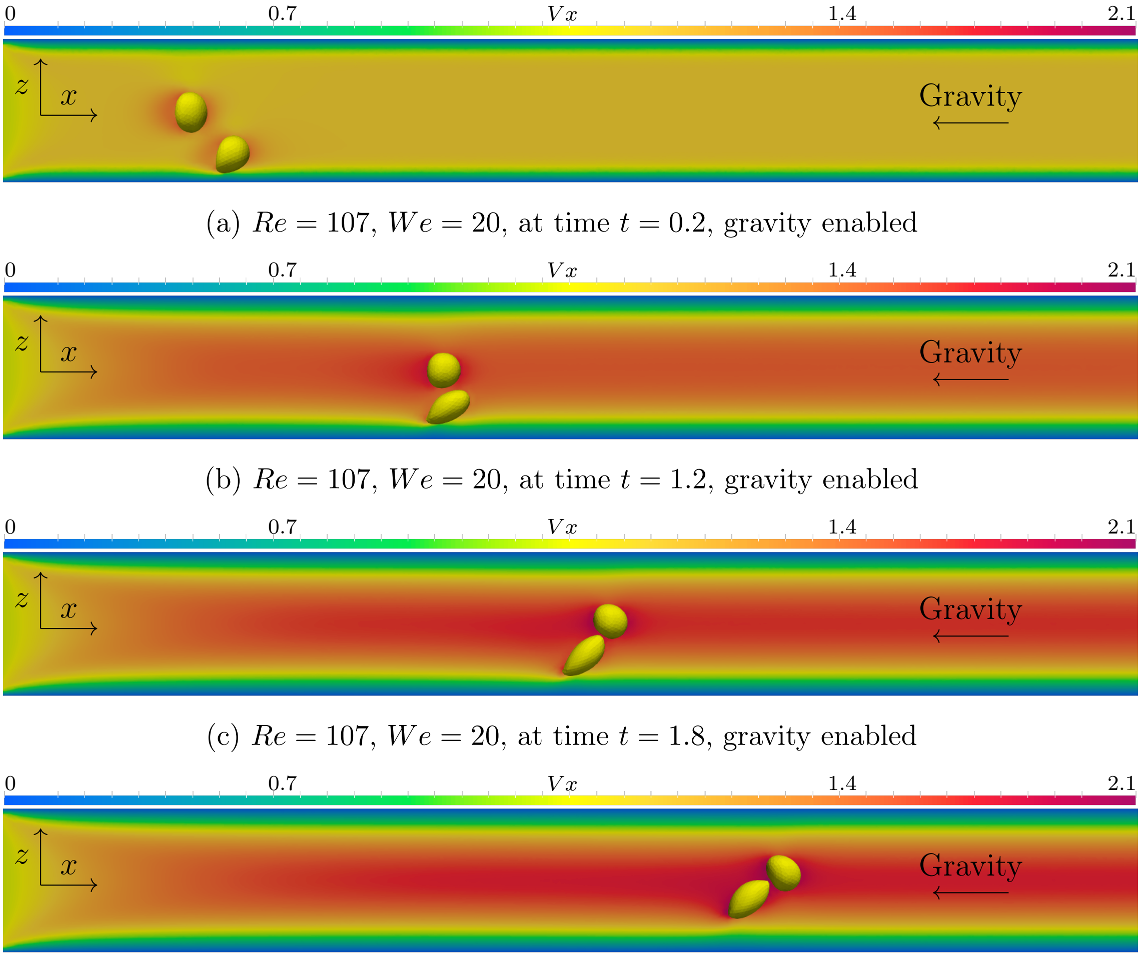
(a) $Re = 107$, $We = 20$, at time $t = 0.2$

(b) $Re = 107$, $We = 20$, at time $t = 1.2$

(c) $Re = 107$, $We = 20$, at time $t = 1.8$

(d) $Re = 107$, $We = 20$, at time $t = 2.4$

Figure 4.29: In a), b), c) and d), the simulation with parameters $Re = 107$ and $We = 20$ is displayed at times $t = 0.2$, $t = 1.2$, $t = 1.8$, and $t = 2.4$. V_x represents the axial velocity component. Gravity forces were disabled, and the simulation was executed over 300 time steps of $dt = 0.01$.



(a) $Re = 107$, $We = 20$, at time $t = 0.2$, gravity enabled

(b) $Re = 107$, $We = 20$, at time $t = 1.2$, gravity enabled

(c) $Re = 107$, $We = 20$, at time $t = 1.8$, gravity enabled

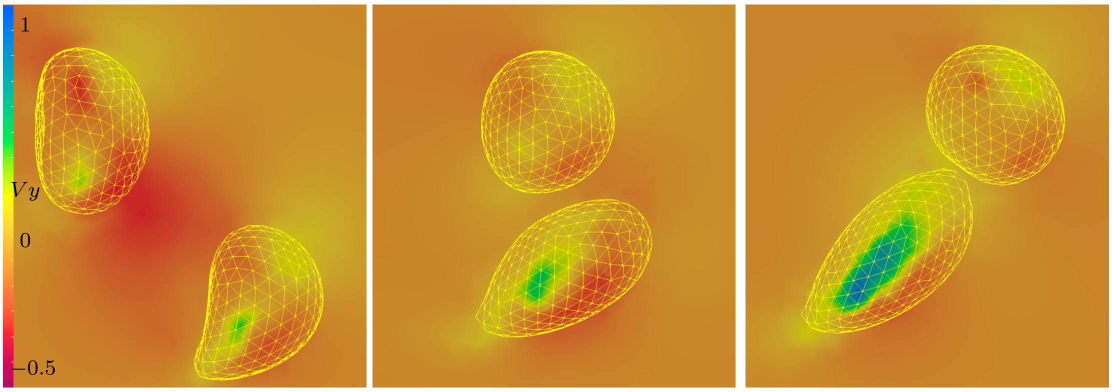
(d) $Re = 107$, $We = 20$, at time $t = 2.4$, gravity enabled

Figure 4.30: In a), b), c) and d), the simulation with parameters $Re = 107$ and $We = 20$ is displayed at times $t = 0.2$, $t = 1.2$, $t = 1.8$, and $t = 2.4$. V_x represents the axial velocity component. Gravity forces, pointing to the left, were active, and the simulation was executed over 300 time steps of $dt = 0.01$.

stretched toward the center, moving into the free bubble's trajectory, and causing a contact. The current two-phase bubble-merging model was disabled for all simulations, since coalescence is not the aim of the present research, but it can be argued that it could occur under these conditions.

Figure 4.34 presents the change in the z-coordinate for both bubbles in all six simulations. In all simulations, some influence can be seen in the plot, but the impact in scenarios with $We = 10$ and $We = 20$ is not as significant. The remaining simulations, with $We = 40$, show some relevant influence on both bubbles' trajectories, especially the simulations with gravity forces.

In fig. 4.35, the x-direction velocity component for both bubbles is presented. It can be inferred from the plot that the free bubble is slowed down by the shear bubble's presence, up until the point where the free bubble overtakes it. It is also evidenced that the shear bubble accelerates after the free bubble's passage, especially



(a) Re = 107, We = 20, at time $t = 0.2$, gravity enabled

(b) Re = 107, We = 20, at time $t = 1.2$, gravity enabled

(c) Re = 107, We = 20, at time $t = 1.8$, gravity enabled

Figure 4.31: Detail near the bubbles presenting the vertical velocity fields for the simulation with parameters $Re = 107$ and $We = 20$ with gravity forces enabled at times a) $t = 0.2$, b) $t = 1.2$, and c) $t = 1.8$. The bubbles are displayed in a wire frame overlaid on the velocity fields. It is possible to notice the increase in the velocity values inside the shear bubble, pushing it toward the channel center.

in the simulations where gravity effects are enabled.

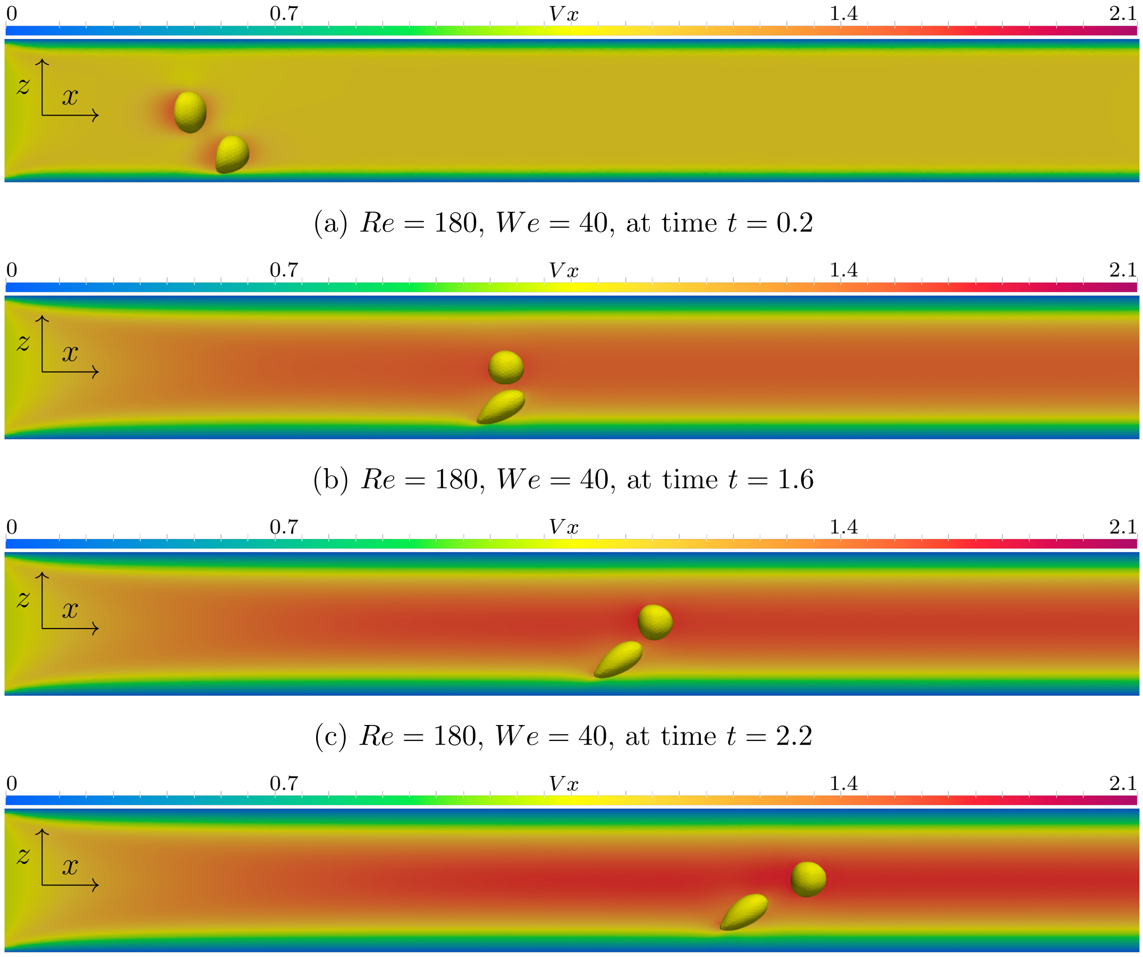
4.3.8 Complex Geometries

In fig. 4.36, additional complex geometries are presented to execute the two-phase flow simulation. These geometries are meant to challenge the simulation and represent possible real-world scenarios.

All simulations were executed with the same fluid properties, $\rho = \mu = 1$, while the bubble fluid properties are $\rho = 0.000892$ and $\mu = 0.0000142$, with gravity disabled. The same time step was selected for all simulations, with a time step of $\Delta t = 0.01$, over a maximum of 400 time steps, however only the expanding channel simulation completed all time steps. The converging channel simulation was interrupted by the bubble reaching the fluid domain's end, and the curved channel simulation was interrupted near the end by a bubble breakup. The same interface thickness $\epsilon = 1.5h_f$ was used for all three simulations.

The converging channel simulation has 1744069 nodes, and was executed with Reynolds number $Re = 100$ and Weber number $We = 10$. The expanding channel simulation domain has 1720930 nodes and was executed with $Re = 10$ and $We = 100$, and the curved channel simulation has 2963087 nodes, and was executed with $Re = 10$ and $We = 10$.

The results of the converging channel simulation can be observed in fig. 4.37. It initiates with the bubble gradually adopting a mushroom shape as it advances toward the channel constriction. Upon reaching the constriction, the bubble undergoes a rapid change in shape, which occurs very rapidly compared to other observed



(a) $Re = 180$, $We = 40$, at time $t = 0.2$

(b) $Re = 180$, $We = 40$, at time $t = 1.6$

(c) $Re = 180$, $We = 40$, at time $t = 2.2$

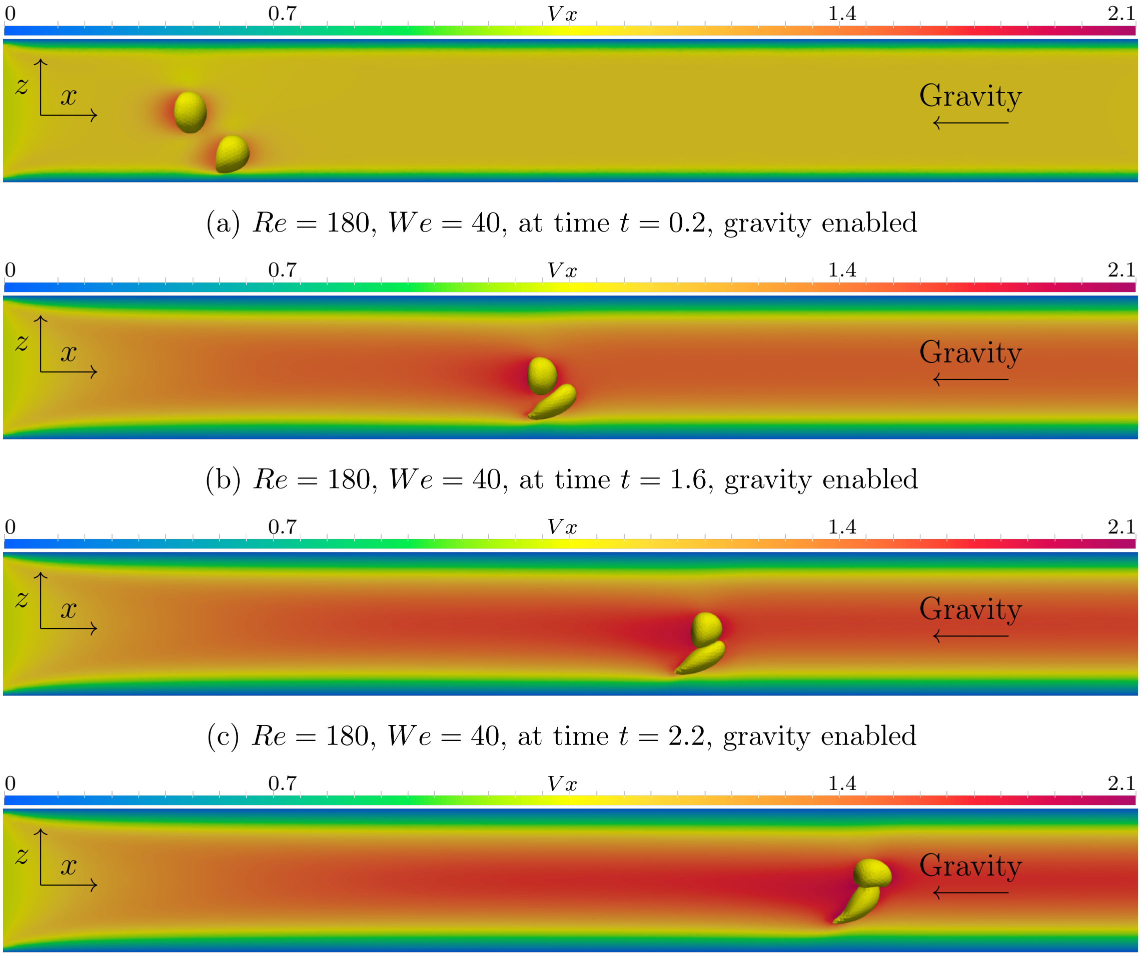
(d) $Re = 180$, $We = 40$, at time $t = 2.8$

Figure 4.32: In a), b), c) and d), the simulation with parameters $Re = 180$ and $We = 40$ is displayed at times $t = 0.2$, $t = 1.6$, $t = 2.2$, and $t = 2.8$. V_x represents the axial velocity component. Gravity forces were disabled, and the simulation was executed over 300 time steps of $dt = 0.01$.

phenomena. In just a few time steps, the triangular shape of the bubble transforms; its top takes on a Taylor bubble shape, while the middle undergoes constriction until it passes through the constriction section assuming the Taylor shape completely.

In the expanding channel simulation presented in fig. 4.38, initially, the bubble swiftly assumes the form of a Taylor bubble and advances towards the expansion section. As the bubble traverses the expansion, it swells, becomes curved, and then the lower portion of the bubble recedes, resulting in a bell-shaped configuration as it exits the channel expansion.

Finally, in the curved channel simulation displayed in fig. 4.39, the bubble begins to take on a mushroom shape, but the velocity fields promptly deform it, pulling its top towards the curve. The bubble starts to hollow out, undergoing a drastic change in shape. As it ascends towards the top of the channel, it leaves a trailing portion behind, gradually distancing itself from it until separation occurs, concluding the



(a) $Re = 180$, $We = 40$, at time $t = 0.2$, gravity enabled

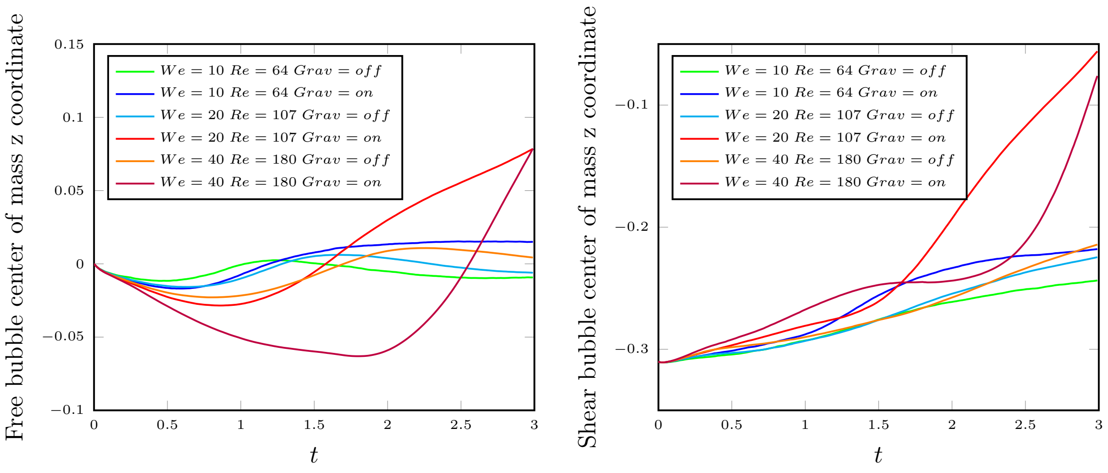
(b) $Re = 180$, $We = 40$, at time $t = 1.6$, gravity enabled

(c) $Re = 180$, $We = 40$, at time $t = 2.2$, gravity enabled

(d) $Re = 180$, $We = 40$, at time $t = 2.8$, gravity enabled

Figure 4.33: In a), b), c) and d), the simulation with parameters $Re = 180$ and $We = 40$ is displayed at times $t = 0.2$, $t = 1.6$, $t = 2.2$, and $t = 2.8$. V_x represents the axial velocity component. Gravity forces, pointing to the left, were active, and the simulation was executed over 300 time steps of $dt = 0.01$. Bubble coalescence may occur under these conditions.

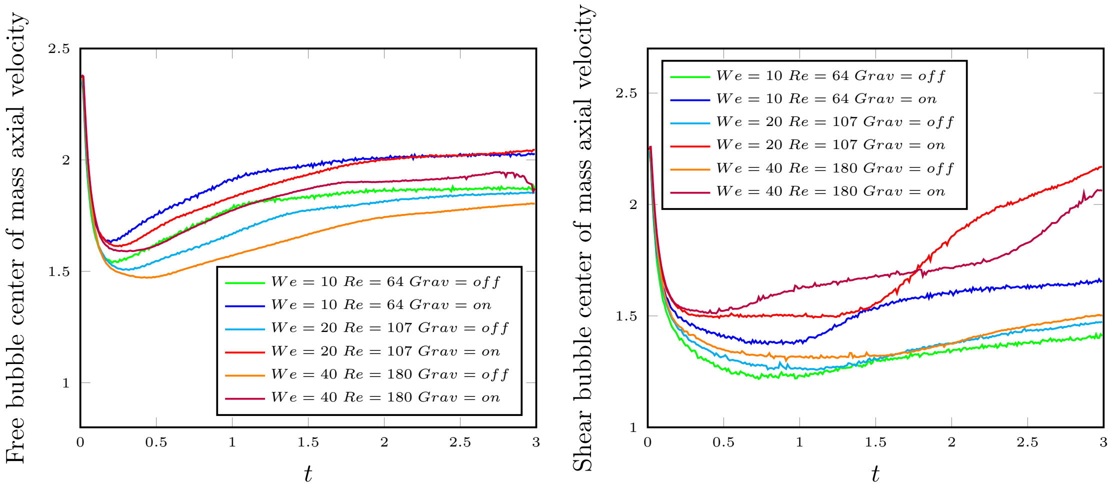
simulation due to the unsupported breakup.



(a) Center of mass z coordinate position of the free bubble

(b) Center of mass z coordinate position of the shear bubble

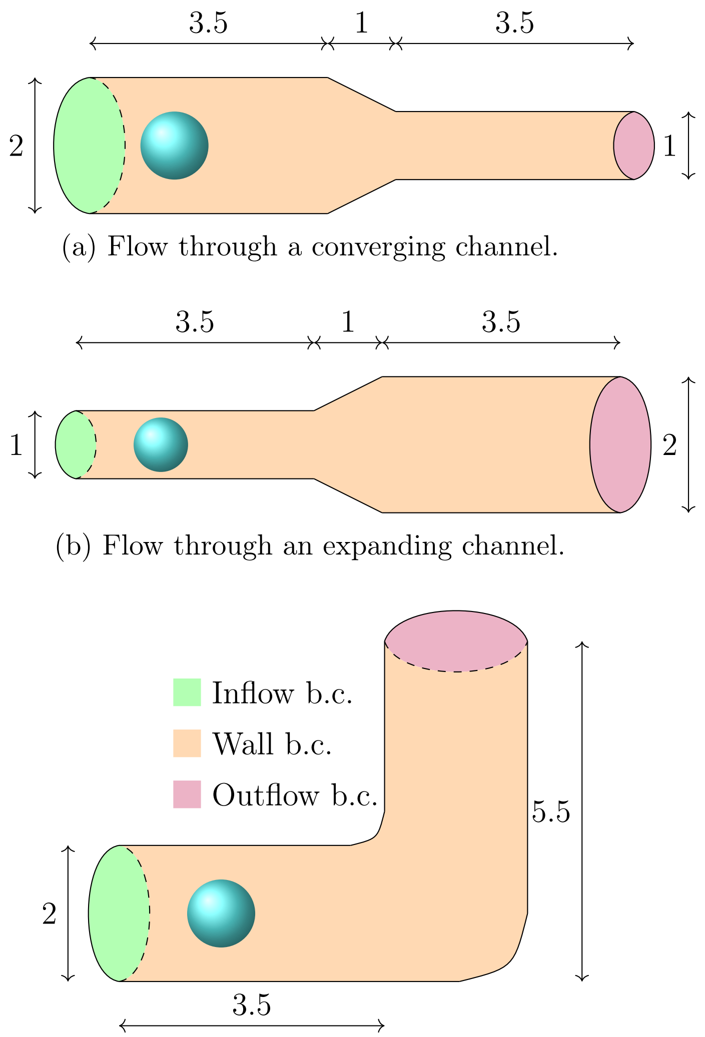
Figure 4.34: In this figure, the change in the z-coordinate of the center-of-mass position is depicted for a) the free bubble and b) the shear bubble as a measure of the influence of the bubbles on each other's trajectory. It can be noticed, especially in the simulations with Reynolds number $Re = 180$ and Weber number $We = 40$, that the shear bubble moves toward the channel's center, and the free bubble is displaced by it.



(a) Free bubble center of mass axial velocity

(b) Shear bubble center of mass axial velocity

Figure 4.35: This figure presents the velocity in the x-direction of a) the free bubble and b) the shear bubble. The moment the free bubble overtakes the shear bubble happens at $t = 0.9$, $t = 1.2$, and $t = 1.6$ in the simulations with Weber numbers $We = 10$, $We = 20$, and $We = 40$.

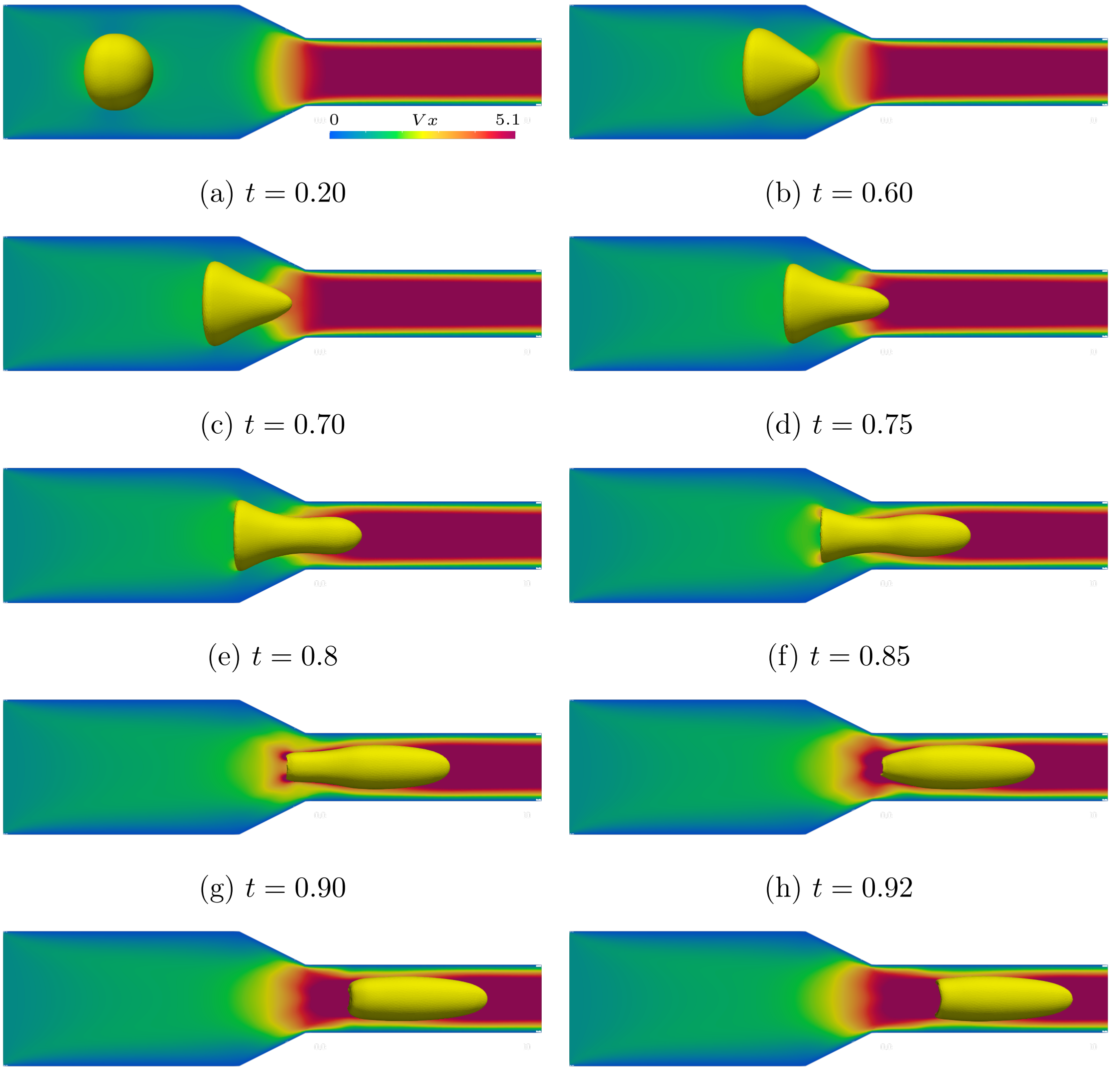


(a) Flow through a converging channel.

(b) Flow through an expanding channel.

(c) Flow through a curved channel.

Figure 4.36: Complex Geometries. In a) and c), the diameter of the bubble is $d_b = 1$. In b), the bubble's diameter is $d_b = 0.8$. The bubble center is positioned at the channel's central axis, at position $x = 1$ in a), b) and c).



(a) $t = 0.20$

(b) $t = 0.60$

(c) $t = 0.70$

(d) $t = 0.75$

(e) $t = 0.8$

(f) $t = 0.85$

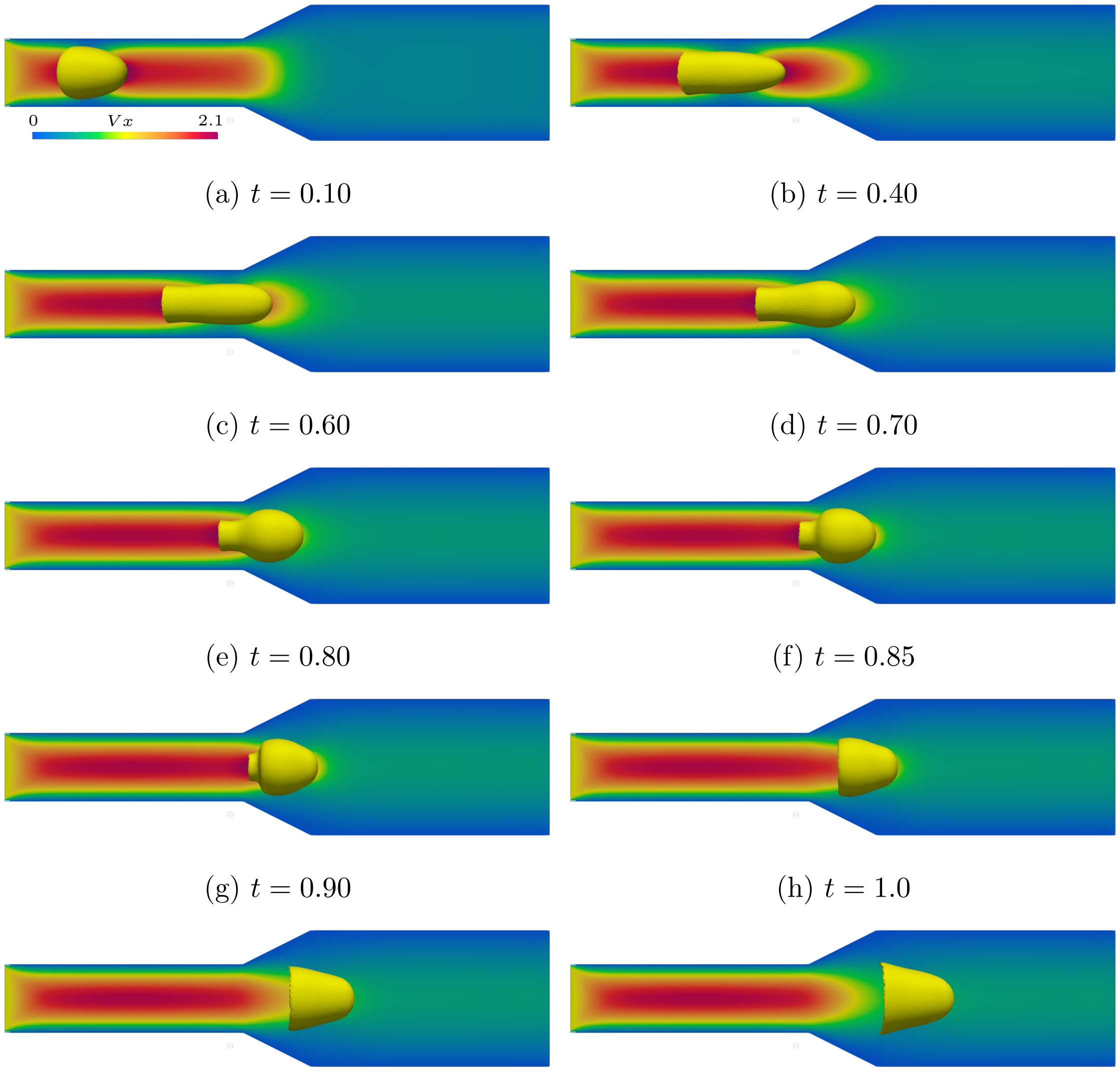
(g) $t = 0.90$

(h) $t = 0.92$

(i) $t = 0.94$

(j) $t = 0.96$

Figure 4.37: Transversal cut of the converging channel simulation results displaying the bubble. In this simulation, channel diameter changes from $d = 2$ to $d = 1$ over a length of $l = 0.5$. The bubble's starting diameter is $d_b = 1$



(a) $t = 0.10$

(b) $t = 0.40$

(c) $t = 0.60$

(d) $t = 0.70$

(e) $t = 0.80$

(f) $t = 0.85$

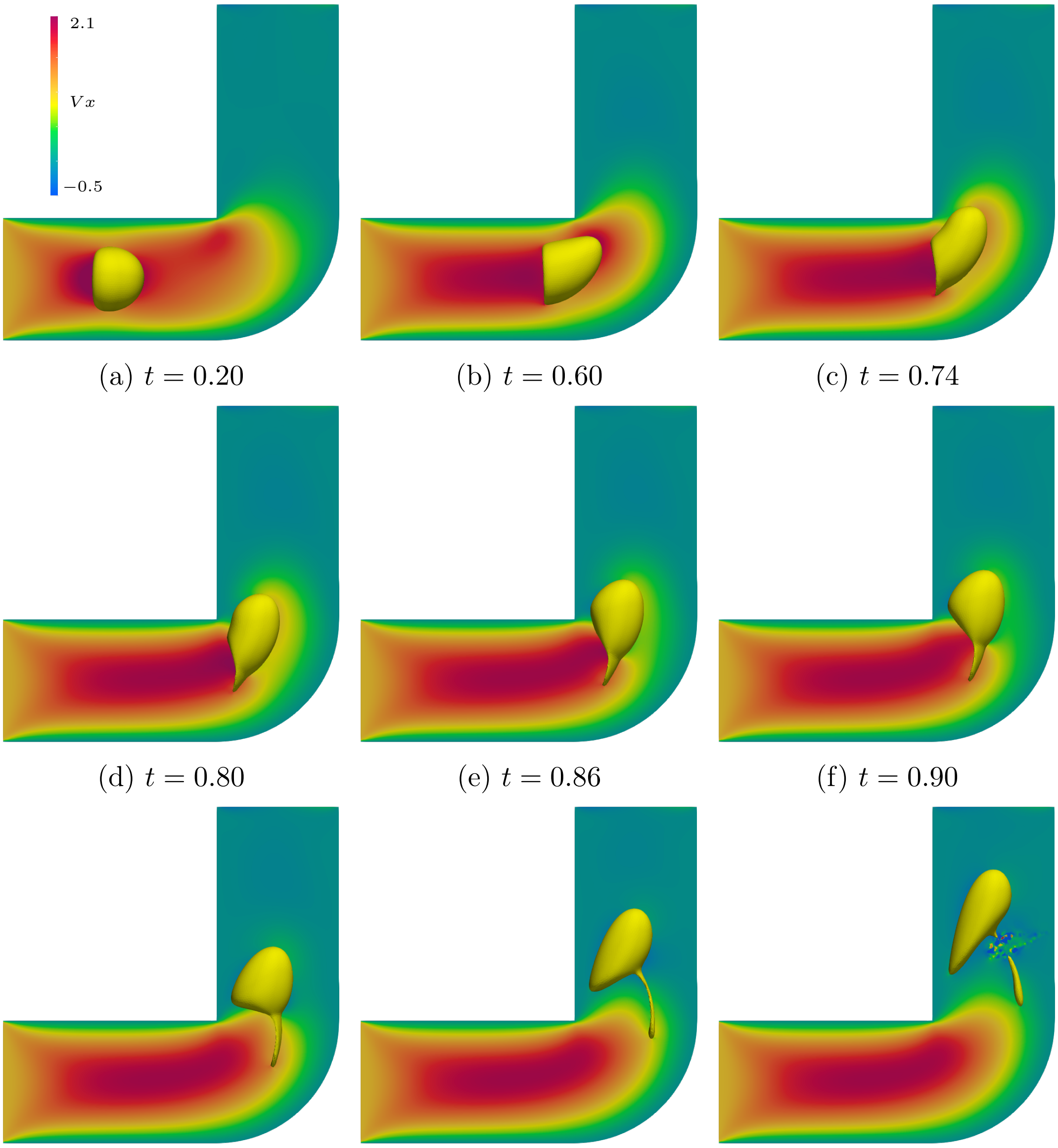
(g) $t = 0.90$

(h) $t = 1.0$

(i) $t = 1.2$

(j) $t = 1.6$

Figure 4.38: Transversal cut of the expanding channel simulation results displaying the bubble. In this simulation, channel diameter changes from $d = 1$ to $d = 2$ over a length of $l = 0.5$. The bubble's starting diameter is $d_b = 0.8$



(a) $t = 0.20$

(b) $t = 0.60$

(c) $t = 0.74$

(d) $t = 0.80$

(e) $t = 0.86$

(f) $t = 0.90$

(g) $t = 1.00$

(h) $t = 1.15$

(i) $t = 1.3$

Figure 4.39: Transversal cut of the curved channel simulation results displaying the bubble. The bubble's starting diameter is $d_b = 1$. A bubble breakup happens at the end of the simulation, displayed in i).

Chapter 5

Conclusion

5.1 Discussion

The goal of the present work was to develop a methodology for studying three-dimensional two-phase flow in complex geometries using the Finite Element method and a Front-Tracking approach to represent the fluid interface. The Navier-Stokes equations were discretized through the one-fluid approach on unstructured meshes. The semi-Lagrangian method was employed to represent the material derivative, and the Laplace-Beltrami operator was utilized to calculate curvature. Applying the aforementioned methodology, a series of single-phase and two-phase tests were conducted, and the simulations' behavior was subsequently analyzed and described.

The single-phase Hagen-Poiseuille test case demonstrated both stability and accuracy, exhibiting convergence toward exact results with mesh refinement. The results of the single-phase microchannel simulations provided valuable insights into optimal microchannel geometries. Circular, square, and hexagonal cross sections yielded predominantly symmetric velocity fields, with circular contours near the center. Notably, the square cross-section displayed a less steep velocity gradient close to the center, suggesting potential advantages for two-phase simulations.

The two-phase static droplet test exhibited low-intensity spurious velocities, consistent with values reported in the literature. Maximum values occurred at the simulation's outset, when there is some adjustment to the droplet's position. Following the initial time steps, there is a significant decrease in the intensity of spurious velocities, reducing by nearly an order of magnitude. This is due to the difference between the initial mesh shape and the numerical code balance between surface tension and pressure. Tests where the fluid mesh was fixed and the interface mesh refined demonstrated a trend ranging between linear and quadratic as the interface mesh is refined, until reaching a threshold where the interface mesh refinement approximates the fluid mesh refinement. Beyond this point, further increases in interface refinement escalated spurious velocities, adversely affecting results. The static droplet test conducted with different fluid and interface meshes revealed an approximately linear relationship between the number of fluid mesh nodes and the intensity of spurious velocities.

The oscillating droplet test case yielded accurate results in both amplitude and oscillation frequency. Nevertheless, it proved to be the most challenging test among all. The accuracy of this test relies heavily on an precise representation of the surface tension force, necessitating a highly refined fluid mesh for satisfactory outcomes. The selected interface thickness influences the oscillation frequency, with smaller thicknesses yielding more accurate results. One possible explanation for this phenomenon is that increasing the interface thickness leads to a larger droplet radius, directly affecting the frequency. Another notable observation is that the chosen time step affects the oscillation amplitude, requiring some trial and error to find the optimal value. Additionally, there is some mesh dependency, where certain interface meshes produce better results than others, even when all other factors remain constant. Fortunately, this dependency diminishes as mesh refinement increases, ensuring that different interface meshes with similar refinement levels yield comparable results.

The results from both the static droplet test and the rising bubble test demonstrate that when the average edge length of the bubble mesh is smaller than that of the fluid mesh, the simulation becomes unstable. While refining the interface mesh does improve curvature calculation and simulation accuracy, this enhancement appears limited by the fluid mesh refinement.

The isolated rising bubble tests exhibited excellent agreement with experimental data. Despite variations in time step and interface thickness, this test proved to be less sensitive to these parameters compared to the oscillating droplet test. The gravitational force involved in this simulation exert far greater influence than the surface tension force, lessening the influence of these parameters.

The Taylor bubble tests demonstrated agreement with experimental data, albeit to a lesser extent than the isolated rising bubble test case. In this scenario, the interface thickness played a more significant role, likely due to the proximity to the walls of the channel.

Finally, the simulation of various complex geometries provided interesting insights into bubble behavior in these specific situations, allowing for informed engineering decisions when designing microchannels or heat transfer devices.

The optimal value for the interface thickness, ϵ, was determined through testing. For most simulations executed, the optimal value seem to be close to $\epsilon = 1.5h_f$. The exception to this was the oscillating droplet test, where lower values provided more accurate oscillation frequency results. Care must also be taken when selecting the ϵ

value in fluid meshes where h_f varies over the domain, depending on how the value of h_f is input or calculated in the simulation.

Overall, the method demonstrates both accuracy and stability. Its implementation is relatively straightforward, and it eliminates time-consuming remeshing procedures for the fluid mesh present in other front-tracking approaches.

5.2 Suggestions

Further research is recommended to enhance the exchange of information between the interface mesh and the fluid mesh. This improvement is expected to yield better-behaved surface tension dominant tests, such as the oscillating droplet, and to a lesser degree, the static droplet.

The implementation of second-order finite elements is also anticipated to enhance the method's accuracy within the same simulation time frame. Although individually more expensive, second-order elements are expected to significantly improve accuracy, especially considering that the central mini element nodes, which constitute the majority of nodes present in the fluid mesh, do not contribute to a more accurate simulation. It should be noted however, that the limitations observed in the curvature calculation might limit accuracy gains.

Lastly, to form a comprehensive computational fluid dynamics package, the simulation of heat and mass transfer can be introduced. While heat transfer simulation is relatively straightforward, the mass transfer technique, though present in similar front-tracking techniques, needs to be adapted to the current uncoupled method.

References

[1] TOM SAENEN, M. B. “Numerical model of a two-phase microchannel heat sink electronics cooling system”, *International Journal of Thermal Sciences*, v. 59, pp. 214–223, 2012. doi: http://dx.doi.org/10.1016/j.ijthermalsci.2012.04.018.

[2] JIA LIU, X. L. “Numerical evaluation on multiphase flow and heat transfer during thermal stimulation enhanced shale gas recovery”, *Applied Thermal Engineering*, v. 178, pp. 155554, 2020. doi: https://doi.org/10.1016/j.applthermaleng.2020.115554.

[3] A. ODUKOYA, G. F. N. “Heat transfer and multiphase flow with hydrate formation in subsea pipelines”, *Heat Mass Transfer*, v. 51, pp. 901–909, 2015. doi: 10.1007/s00231-014-1457-3.

[4] H.U. HELVACI, Z. K. “Mathematical modelling and simulation of multiphase flow in a flat plate solar energy collector”, *Energy Conversion and Management*, v. 106, pp. 139–150, 2015. doi: http://dx.doi.org/10.1016/j.enconman.2015.09.028.

[5] RADMAN, S., FIORINA, C., PAUTZ, A. “Development of a novel two-phase flow solver for nuclear reactor analysis: Validation against sodium boiling experiments”, *Nuclear Engineering and Design*, v. 384, pp. 111422, dec. 2021. ISSN: 00295493. doi: 10.1016/j.nucengdes.2021.111422. Available at: <`https://linkinghub.elsevier.com/retrieve/pii/S0029549321003745`>.

[6] CHIRAG R. KHARANGATE, I. M. “Review of computational studies on boiling and condensation”, *International Journal of Heat and Mass Transfer*, v. 108, pp. 1164–1196, 2017. doi: http://dx.doi.org/10.1016/j.ijheatmasstransfer.2016.12.065.

[7] BAIG, T., REHMAN, Z., TARIQ, H. A., et al. “Thermal Performance Investigation of Slotted Fin Minichannel Heat Sink for Microprocessor

Cooling", *Energies*, v. 14, n. 19, pp. 6347, oct. 2021. ISSN: 1996-1073. doi: 10.3390/en14196347. Available at: <https://www.mdpi.com/1996-1073/14/19/6347>.

[8] PARK, M.-C., MA, S.-B., KIM, K.-Y. "Optimization of a Wavy Microchannel Heat Sink with Grooves", *Processes*, v. 9, n. 2, pp. 373, feb. 2021. ISSN: 2227-9717. doi: 10.3390/pr9020373. Available at: <https://www.mdpi.com/2227-9717/9/2/373>.

[9] TARIQ, H. A., ANWAR, M., MALIK, A., et al. "Hydro-thermal performance of normal-channel facile heat sink using TiO2-H2O mixture (Rutile–Anatase) nanofluids for microprocessor cooling", *Journal of Thermal Analysis and Calorimetry*, v. 145, n. 5, pp. 2487–2502, sep. 2021. ISSN: 1388-6150, 1588-2926. doi: 10.1007/s10973-020-09838-x. Available at: <https://link.springer.com/10.1007/s10973-020-09838-x>.

[10] EL-KHOULY, M., EL BOUZ, M., SULTAN, G. "Experimental and computational study of using nanofluid for thermal management of electronic chips", *Journal of Energy Storage*, v. 39, pp. 102630, jul. 2021. ISSN: 2352152X. doi: 10.1016/j.est.2021.102630. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S2352152X21003728>.

[11] ANISH, M., BENCY, P., JAYAPRAKASH, V., et al. "An Experimental Assessment into the Pressure and Thermal Efficacy of Chemically Synthesized Nanofluids in Computer Cooling Applications", *Journal of Nanomaterials*, v. 2022, pp. 1–16, nov. 2022. ISSN: 1687-4129, 1687-4110. doi: 10.1155/2022/4647759. Available at: <https://www.hindawi.com/journals/jnm/2022/4647759/>.

[12] BARRUBEEAH, M., RADY, M., ATTAR, A., et al. "Design, modeling and parametric optimization of thermoelectric cooling systems for high power density electronic devices", *International Journal of Low-Carbon Technologies*, v. 16, n. 3, pp. 1060–1076, sep. 2021. ISSN: 1748-1325. doi: 10.1093/ijlct/ctab032. Available at: <https://academic.oup.com/ijlct/article/16/3/1060/6248809>.

[13] DE MEY, G., KOS, A., GÓRECKI, K. "Optimal Temperature Regulation of Integrated Circuits with Peltier Heat Pumps", *Energies*, v. 15, n. 3, pp. 1125, feb. 2022. ISSN: 1996-1073. doi: 10.3390/en15031125. Available at: <https://www.mdpi.com/1996-1073/15/3/1125>.

[14] ANJOS, G. "Moving mesh methods for two-phase flow systems: Assessment, comparison and analysis", *Computers & Fluids*, v. 228,

pp. 105053, oct. 2021. ISSN: 00457930. doi: 10.1016/j.compfluid.2021.105053. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0045793021002176>.

[15] ANJOS, G. R., OLIVEIRA, G. P., MANGIAVACCHI, N., et al. "One- and two-step semi-Lagrangian integrators for arbitrary Lagrangian–Eulerian-finite element two-phase flow simulations", *International Journal for Numerical Methods in Fluids*, v. 94, n. 6, pp. 632–654, jun. 2022. ISSN: 0271-2091, 1097-0363. doi: 10.1002/fld.5069. Available at: <https://onlinelibrary.wiley.com/doi/10.1002/fld.5069>.

[16] ZIENKIEWICS O. C., C. Y. K. "Finite elements in the solution of field problems", *The Engineer*, , n. 517-10, 1965.

[17] ERGATOUDIS, I., IRONS, B., ZIENKIEWICZ, O. "Curved, isoparametric, "quadrilateral" elements for finite element analysis", *International Journal of Solids and Structures*, v. 4, n. 1, pp. 31–42, jan. 1968. ISSN: 00207683. doi: 10.1016/0020-7683(68)90031-0. Available at: <https://linkinghub.elsevier.com/retrieve/pii/0020768368900310>.

[18] C. W. HIRT, B. D. N. "Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries", *Journal of Computational Physics*, v. 39, pp. 221–225, 1981.

[19] LYU, S., WANG, K., ZHANG, Z., et al. "A hybrid VOF-IBM method for the simulation of freezing liquid films and freezing drops", *Journal of Computational Physics*, v. 432, pp. 110160, may 2021. ISSN: 00219991. doi: 10.1016/j.jcp.2021.110160. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0021999121000528>.

[20] ISSAKHOV, A., IMANBERDIYEVA, M. "Numerical simulation of the movement of water surface of dam break flow by VOF methods for various obstacles", *International Journal of Heat and Mass Transfer*, v. 136, pp. 1030–1051, jun. 2019. ISSN: 00179310. doi: 10.1016/j.ijheatmasstransfer.2019.03.034. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0017931018349287>.

[21] POZZETTI, G., PETERS, B. "A multiscale DEM-VOF method for the simulation of three-phase flows", *International Journal of Multiphase Flow*, v. 99, pp. 186–204, feb. 2018. ISSN: 03019322. doi: 10.1016/j.ijmultiphaseflow.2017.10.008. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0301932216307327>.

[22] GAROOSI, F., MERABTENE, T., MAHDI, T.-F. “Numerical simulation of merging of two rising bubbles with different densities and diameters using an enhanced Volume-Of-Fluid (VOF) model”, *Ocean Engineering*, v. 247, pp. 110711, mar. 2022. ISSN: 00298018. doi: 10.1016/j.oceaneng.2022.110711. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0029801822001639>.

[23] OSHER, S., SETHIAN, J. A. “Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations”, *Journal of Computational Physics*, v. 79, n. 1, pp. 12–49, nov. 1988. ISSN: 00219991. doi: 10.1016/0021-9991(88)90002-2. Available at: <https://linkinghub.elsevier.com/retrieve/pii/0021999188900022>.

[24] YAP, Y., LI, H., LOU, J., et al. “Numerical modeling of three-phase flow with phase change using the level-set method”, *International Journal of Heat and Mass Transfer*, v. 115, pp. 730–740, dec. 2017. ISSN: 00179310. doi: 10.1016/j.ijheatmasstransfer.2017.08.076. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0017931017317933>.

[25] FREY, P., KAZERANI, D., TA, T. “An adaptive numerical scheme for solving incompressible 2-phase and free-surface flows: Numerical scheme for incompressible 2-phase and free-surface flows”, *International Journal for Numerical Methods in Fluids*, v. 87, n. 11, pp. 543–582, aug. 2018. ISSN: 02712091. doi: 10.1002/fld.4502. Available at: <https://onlinelibrary.wiley.com/doi/10.1002/fld.4502>.

[26] ZHANG, Y., ZOU, Q., GREAVES, D. “Numerical simulation of free-surface flow using the level-set method with global mass correction”, *International Journal for Numerical Methods in Fluids*, pp. n/a–n/a, 2009. ISSN: 02712091, 10970363. doi: 10.1002/fld.2090. Available at: <https://onlinelibrary.wiley.com/doi/10.1002/fld.2090>.

[27] SUSSMAN, M., PUCKETT, E. G. “A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows”, *Journal of Computational Physics*, v. 162, n. 2, pp. 301–337, aug. 2000. ISSN: 00219991. doi: 10.1006/jcph.2000.6537. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0021999100965379>.

[28] ANDREA FERRARI, MIRCO MAGNINI, J. R. T. “A Flexible Coupled Level Set and Volume of Fluid (flexCLV) method to simulate microscale two-phase flow in non-uniform and unstructured meshes”, *International Jour-*

nal of Multiphase Flow, v. 91, pp. 276–295, 2017. doi: http://dx.doi.org/10.1016/j.ijmultiphaseflow.2017.01.017.

[29] D.L. SUN, W. T. "A coupled volume-of-fluid and level set (VOSET) method for computing incompressible two-phase flows", *International Journal of Heat and Mass Transfer*, v. 53, pp. 645–655, 2010. doi: http://dx.doi.org/10.1016/j.ijheatmasstransfer.2009.10.030.

[30] KONG LING, Z.-H. L. "A three-dimensional volume of fluid I& level set (VOSET) method for incompressible two-phase flow", *Computers I& Fluids*, v. 118, pp. 293–304, 2015. doi: http://dx.doi.org/10.1016/j.compfluid.2015.06.018.

[31] FEDKIW, R. P. "Coupling an Eulerian Fluid Calculation to a Lagrangian Solid Calculation with the Ghost Fluid Method", *Journal of Computational Physics*, v. 175, n. 1, pp. 200–224, jan. 2002. ISSN: 00219991. doi: 10.1006/jcph.2001.6935. Available at: <`https://linkinghub.elsevier.com/retrieve/pii/S0021999101969359`>.

[32] UNVERDI, S. O., TRYGGVASON, G. "A front-tracking method for viscous, incompressible, multi-fluid flows", *Journal of Computational Physics*, v. 100, n. 1, pp. 25–37, may 1992. ISSN: 00219991. doi: 10.1016/0021-9991(92)90307-K. Available at: <`https://linkinghub.elsevier.com/retrieve/pii/002199919290307K`>.

[33] HUA, J., MORTENSEN, D. "A front tracking method for simulation of two-phase interfacial flows on adaptive unstructured meshes for complex geometries", *International Journal of Multiphase Flow*, v. 119, pp. 166–179, oct. 2019. ISSN: 03019322. doi: 10.1016/j.ijmultiphaseflow.2019.07.011. Available at: <`https://linkinghub.elsevier.com/retrieve/pii/S030193221930062X`>.

[34] IRFAN, M., MURADOGLU, M. "A front tracking method for direct numerical simulation of evaporation process in a multiphase system", *Journal of Computational Physics*, v. 337, pp. 132–153, may 2017. ISSN: 00219991. doi: 10.1016/j.jcp.2017.02.036. Available at: <`https://linkinghub.elsevier.com/retrieve/pii/S0021999117301304`>.

[35] E. GROS, G. ANJOS, J. T. "Moving mesh method for direct numerical simulation of two-phase flow with phase change", *Applied Mathematics and Computation*, v. 339, pp. 636–650, 2018. doi: https://doi.org/10.1016/j.amc.2018.07.052.

[36] YU ZHANG, A. C. “A locally discontinuous ALE finite element formulation for compressible phase change problems”, *Journal of Computational Physics*, v. 393, pp. 438–464, 2019. doi: https://doi.org/10.1016/j.jcp.2019.04.039.

[37] ANJOS, G., MANGIAVACCHI, N., THOME, J. “An ALE-FE method for two-phase flows with dynamic boundaries”, *Computer Methods in Applied Mechanics and Engineering*, v. 362, pp. 112820, 2020. ISSN: 0045-7825. doi: https://doi.org/10.1016/j.cma.2020.112820. Available at: <`http://www.sciencedirect.com/science/article/pii/S0045782520300013`>.

[38] ZEKANG CHENG, JIE LI, C. Y. L. “An exactly force-balanced boundary-conforming arbitrary-Lagrangian-Eulerian method for interfacial dynamics”, *Journal of Computational Physics*, v. 408, pp. 109237, 2020. doi: https://doi.org/10.1016/j.jcp.2020.109237.

[39] KANG-XIN CHEN, HAO SHI, H.-F. S. “Finite element modeling of macrosegregation coupled with shrinkage cavity in steel ingots using arbitrary Lagrangian-Eulerian model”, *China Foundry*, v. 16, 2019. doi: https://doi.org/10.1007/s41230-019-9061-2.

[40] CAGINALP, G. “An analysis of a phase field model of a free boundary”, *Archive for Rational Mechanics and Analysis*, v. 92, n. 3, pp. 205–245, sep. 1986. ISSN: 0003-9527, 1432-0673. doi: 10.1007/BF00254827. Available at: <`http://link.springer.com/10.1007/BF00254827`>.

[41] LANGER, J. S. “MODELS OF PATTERN FORMATION IN FIRST-ORDER PHASE TRANSITIONS”. In: *Series on Directions in Condensed Matter Physics*, v. 1, WORLD SCIENTIFIC, pp. 165–186, aug. 1986. ISBN: 9789971978426 9789814415309. doi: 10.1142/9789814415309_0005. Available at: <`http://www.worldscientific.com/doi/abs/10.1142/9789814415309_0005`>.

[42] YANG, J., KIM, J. “A phase-field model and its efficient numerical method for two-phase flows on arbitrarily curved surfaces in 3D space”, *Computer Methods in Applied Mechanics and Engineering*, v. 372, pp. 113382, dec. 2020. ISSN: 00457825. doi: 10.1016/j.cma.2020.113382. Available at: <`https://linkinghub.elsevier.com/retrieve/pii/S0045782520305673`>.

[43] OUARDGHI, A., EL-AMRANI, M., SEAID, M. “An adaptive enriched semi-Lagrangian finite element method for coupled flow-transport problems”, *Computers & Fluids*, v. 240, pp. 105474, may 2022. ISSN: 00457930. doi:

10.1016/j.compfluid.2022.105474. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0045793022001220>.

[44] LIN, X., WU, J., ZHANG, T. “A mesh-free radial basis function–based semi-Lagrangian lattice Boltzmann method for incompressible flows”, *International Journal for Numerical Methods in Fluids*, v. 91, n. 4, pp. 198–211, oct. 2019. ISSN: 0271-2091, 1097-0363. doi: 10.1002/fld.4749. Available at: <https://onlinelibrary.wiley.com/doi/10.1002/fld.4749>.

[45] WILDE, D., KRÄMER, A., REITH, D., et al. “High-order semi-Lagrangian kinetic scheme for compressible turbulence”, *Physical Review E*, v. 104, n. 2, pp. 025301, aug. 2021. ISSN: 2470-0045, 2470-0053. doi: 10.1103/PhysRevE.104.025301. Available at: <https://link.aps.org/doi/10.1103/PhysRevE.104.025301>.

[46] TAVELLI, M., BOSCHERI, W., STRADIOTTI, G., et al. “A mass-conservative semi-implicit volume of fluid method for the Navier–Stokes equations with high order semi-Lagrangian advection scheme”, *Computers & Fluids*, v. 240, pp. 105443, may 2022. ISSN: 00457930. doi: 10.1016/j.compfluid.2022.105443. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0045793022000962>.

[47] BATCHELOR, G. K., YOUNG, A. D. “*An Introduction to Fluid Mechanics*”, *Journal of Applied Mechanics*, v. 35, n. 3, pp. 624–624, sep. 1968. ISSN: 0021-8936, 1528-9036. doi: 10.1115/1.3601282. Available at: <https://asmedigitalcollection.asme.org/appliedmechanics/article/35/3/624/385493/An-Introduction-to-Fluid-Mechanics>.

[48] KUNDU, P. K., COHEN, I. M., DOWLING, D. R. *Fluid mechanics.* 5th ed ed. Waltham, MA, Academic Press, 2012. ISBN: 9780123821003. OCLC: ocn713567396.

[49] J. U. BRACKBILL, D. B. KOTHE, C. Z. “A Continuum Method for Modelling Surface Tension”, *Journal of Computational Physics*, v. 100, pp. 335–354, 1992.

[50] REDDY, J. N. *An introduction to the finite element method.* McGraw-Hill series in mechanical engineering. 3rd ed ed. New York, NY, McGraw-Hill Higher Education, 2006. ISBN: 9780072466850.

[51] O. C. ZIENKIEWICS, R. L. TAYLOR, P. N. *The Finite Element Method for Fluid Dynamics.* 7 ed. The Boulevard, Langford Lane, Kidlington, Oxford, OX5 1GB 225 Wyman Street, Waltham, MA 02451, USA, Elsevier, 2014.

[52] HUGHES, T. J. R. *Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. New York, NY, Dover Publications, 2012. ISBN: 9781306348782. OCLC: 868270692.

[53] ULRICH DIERKES, STEFAN HILDEBRANDT, F. S. *Minimal Surfaces (I)*. 2 ed. Germany, Springer, 2010.

[54] SUSSMAN, M., SMEREKA, P., OSHER, S. "A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow", *Journal of Computational Physics*, v. 114, n. 1, pp. 146–159, sep. 1994. ISSN: 00219991. doi: 10.1006/jcph.1994.1155. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0021999184711557>.

[55] JEAN DONEA, A. H. *Finite Element Methods for Flow Problems*. 1 ed. The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ England, John Wiley & Sons Ltd, 2003.

[56] ANDRÉ ROBERT, TAI LOY LEE, H. R. "A Semi-Lagrangian and Semi-Implicit Numerical Integration Scheme for Multilevel Atmospheric Models", *Monthly Weather Review*, v. 113, pp. 388, 1984.

[57] BOFFI, D., BREZZI, F., FORTIN, M. *Mixed Finite Element Methods and Applications*, v. 44, *Springer Series in Computational Mathematics*. Berlin, Heidelberg, Springer Berlin Heidelberg, 2013. ISBN: 9783642365188 9783642365195. doi: 10.1007/978-3-642-36519-5. Available at: <http://link.springer.com/10.1007/978-3-642-36519-5>.

[58] TEZDUYAR, T., MITTAL, S., RAY, S., et al. "Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements", *Computer Methods in Applied Mechanics and Engineering*, v. 95, n. 2, pp. 221–242, mar. 1992. ISSN: 00457825. doi: 10.1016/0045-7825(92)90141-6. Available at: <https://linkinghub.elsevier.com/retrieve/pii/0045782592901416>.

[59] DOHRMANN, C. R., BOCHEV, P. B. "A stabilized finite element method for the Stokes problem based on polynomial pressure projections", *International Journal for Numerical Methods in Fluids*, v. 46, n. 2, pp. 183–201, sep. 2004. ISSN: 0271-2091, 1097-0363. doi: 10.1002/fld.752. Available at: <https://onlinelibrary.wiley.com/doi/10.1002/fld.752>.

[60] D. N. ARNOLD, F. BREZZI, M. F. "A Stable Finite Element For The Stokes Equations", *Estrato da Calcolo*, v. XXI, n. IV, pp. 337–344, oct 1984.

[61] SHEWCHUK, R. *Delaunay Refinement Mesh Generation.* Delaunay Refinement Mesh Generation, Carnegie Mellon University, Pittsburgh, may 1997.

[62] HIRSCH, C. *Numerical computation of internal and external flows: fundamentals of computational fluid dynamics.* 2nd ed ed. Oxford ; Burlington, MA, Elsevier/Butterworth-Heinemann, 2007. ISBN: 9780750665940. OCLC: ocn148277909.

[63] HYSING, S. "A new implicit surface tension implementation for interfacial flows", *International Journal for Numerical Methods in Fluids*, v. 51, n. 6, pp. 659–672, jun. 2006. ISSN: 02712091. doi: 10.1002/fld.1147. Available at: <`https://onlinelibrary.wiley.com/doi/10.1002/fld.1147`>.

[64] FORTUNA, A. D. O. *Tecnicas computacionais para dinamica dos fluidos: conceitos basicos e aplicacoes.* Sao Paulo, Edusp, 2000. ISBN: 9788531405266. OCLC: 46745875.

[65] DOS ANJOS, G. R. *A 3D ALE Finite Element Method for Two-Phase Flows with Phase Change.* Phd thesis, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, Suisse, 2012.

[66] MARK MEYER, MATHIEU DESBRUN, P. S., BARR, A. "Discrete Differential-Geometry Operators for Triangulated 2-Manifolds", *Visualization and Mathematics III*, v. 3, pp. 35–57, 2003.

[67] CHRISTOPHE GEUZAINE, J.-F. R. "Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities", *International Journal for Numerical Methods in Engineering*, v. 79, n. 11, pp. 1309–1331, 1984.

[68] SIEGER, D., BOTSCH, M. "The Polygon Mesh Processing Library". 2019. http://www.pmp-library.org.

[69] DUNYACH, M., VANDERHAEGHE, D., BARTHE, L., et al. "Adaptive Remeshing for Real-Time Mesh Deformation". In: Otaduy, M.-A., Sorkine, O. (Eds.), *Eurographics 2013 - Short Papers*. The Eurographics Association, 2013. doi: 10.2312/conf/EG2013/short/029-032.

[70] BOTSCH, M., KOBBELT, L. "A remeshing approach to multiresolution modeling". In: *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp. 185–192, Nice France, jul. 2004. ACM. ISBN: 9783905673135. doi: 10.1145/1057432.1057457. Available at: <`https://dl.acm.org/doi/10.1145/1057432.1057457`>.

[71] PAIGE, C. C., SAUNDERS, M. A. "Solution of Sparse Indefinite Systems of Linear Equations", *SIAM Journal on Numerical Analysis*, v. 12, n. 4, pp. 617–629, sep. 1975. ISSN: 0036-1429, 1095-7170. doi: 10.1137/0712047. Available at: <http://epubs.siam.org/doi/10.1137/0712047>.

[72] CHANG, W., GIRALDO, F., PEROT, B. "Analysis of an Exact Fractional Step Method", *Journal of Computational Physics*, v. 180, n. 1, pp. 183–199, jul. 2002. ISSN: 00219991. doi: 10.1006/jcph.2002.7087. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0021999102970877>.

[73] PROSPERETTI, A. "Free oscillations of drops and bubbles: the initial-value problem", *Journal of Fluid Mechanics*, v. 100, n. 02, pp. 333, sep. 1980. ISSN: 0022-1120, 1469-7645. doi: 10.1017/S0022112080001188. Available at: <http://www.journals.cambridge.org/abstract_S0022112080001188>.

[74] DE SOUSA, F., MANGIAVACCHI, N., NONATO, L., et al. "A front-tracking/front-capturing method for the simulation of 3D multi-fluid flows with free surfaces", *Journal of Computational Physics*, v. 198, n. 2, pp. 469–499, aug. 2004. ISSN: 00219991. doi: 10.1016/j.jcp.2004.01.032. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0021999104000488>.

[75] BHAGA, D., WEBER, M. E. "Bubbles in viscous liquids: shapes, wakes and velocities", *Journal of Fluid Mechanics*, v. 105, n. 1, pp. 61, apr. 1981. ISSN: 0022-1120, 1469-7645.

[76] WHITE, E., BEARDMORE, R. "The velocity of rise of single cylindrical air bubbles through liquids contained in vertical tubes", *Chemical Engineering Science*, v. 17, n. 5, pp. 351–361, 1962. ISSN: 00092509.

[77] BROWN, R. A. S. "The mechanics of large gas bubbles in tubes: I. Bubble velocities in stagnant liquids", *The Canadian Journal of Chemical Engineering*, v. 43, n. 5, pp. 217–223, oct. 1965. ISSN: 00084034, 1939019X. doi: 10.1002/cjce.5450430501. Available at: <https://onlinelibrary.wiley.com/doi/10.1002/cjce.5450430501>.

Appendix A

Previous Publications

A.1 Published Journal Papers

- **SANTOS, D. B. V.** ; ANJOS, G. R. ; OLIVEIRA, G. P. ; MANGIAVACCHI, N. ; VALLURI, P. . Numerical Investigation of Gas Bubble Interaction in a Circular Cross-Section Channel in Shear Flow Fluids 2024, 9(2), 32.
- **SANTOS, D. B. V.** ; BANDEIRA, A. A. . NUMERICAL MODELING OF CONTACT PROBLEMS WITH THE FINITE ELEMENT METHOD UTILIZING A B-SPLINE SURFACE FOR CONTACT SURFACE SMOOTHING Latin American Journal of Solids and Structures, 15(8), e77, 2018.

A.2 Submitted

- **SANTOS, D. B. V.** ; VALLURI, P.; ANJOS, G. R. . A Three Dimensional Method for Simulating Interfacial Flow in Complex Geometry. Submetido à: Journal of Computational Physics.
- **VIDAL, R.**; SANTOS, D. B. V.; VALLURI, P.; ANJOS, G. R. . Computation of Deformable Interface Two-Phase Flows: A Semi-Lagrangian Finite Element Approach. Submetido à: International Journal for Numerical Methods in Fluids.

A.3 Published Conference Papers

- **SANTOS, D. B. V.** ; ANJOS, G. R.. THREE DIMENSIONAL FINITE ELEMENT TWO-PHASE FLOW SIMULATION USING A FRONT TRACKING METHOD In: 27th COBEM – International Congress of Mechanical Engineering, 2023. Anais do 27th International Congress of Mechanical Engineering, 2023.

- **SANTOS, D. B. V.** ; ANJOS, GUSTAVO R. . AXISYMMETRIC TWO-PHASE FINITE ELEMENT SIMULATION USING A FRONT TRACKING METHOD ON AN UNSTRUCTURED MESH. In: 19th Brazilian Congress of Thermal Sciences and Engineering, 2022, Bento Gonçalves. Anais do 19th Brazilian Congress of Thermal Sciences and Engineering, 2022.

- **SANTOS, D. B. V.** ; ANJOS, G. R. . FINITE ELEMENT SIMULATION FOR TWO-PHASE FLOW WITH A DECOUPLED FLUID INTERFACE. In: 26th COBEM – International Congress of Mechanical Engineering, 2021. Anais do 26th International Congress of Mechanical Engineering, 2021.

- **SANTOS, D. B. V.** ; ANJOS, G.R. . DECOUPLED MESH METHOD FOR FINITE ELEMENT SIMULATION OF TWO-PHASE SYSTEMS. In: 8th International Symposium on ADVANCES IN COMPUTATIONAL HEAT TRANSFER – CHT-21, 2021, Rio de Janeiro. Proceedings of 8th International Symposium on ADVANCES IN COMPUTATIONAL HEAT TRANSFER – CHT-21, 2021.

- **SANTOS, D. B. V.** ; ANJOS, G. R. . NUMERICAL SIMULATION OF MULTIPHASE FLOWS IN THE CONTEXT OF COVID-19/SARS-CoV-2. In: 18th Brazilian Congress of Thermal Sciences and Engineering, 2020, Bento Gonçalves. Anais do 18th Brazilian Congress of Thermal Sciences and Engineering, 2020.