Universidade do Estado do Rio de Janeiro

Centro de Ciência e Tecnologia
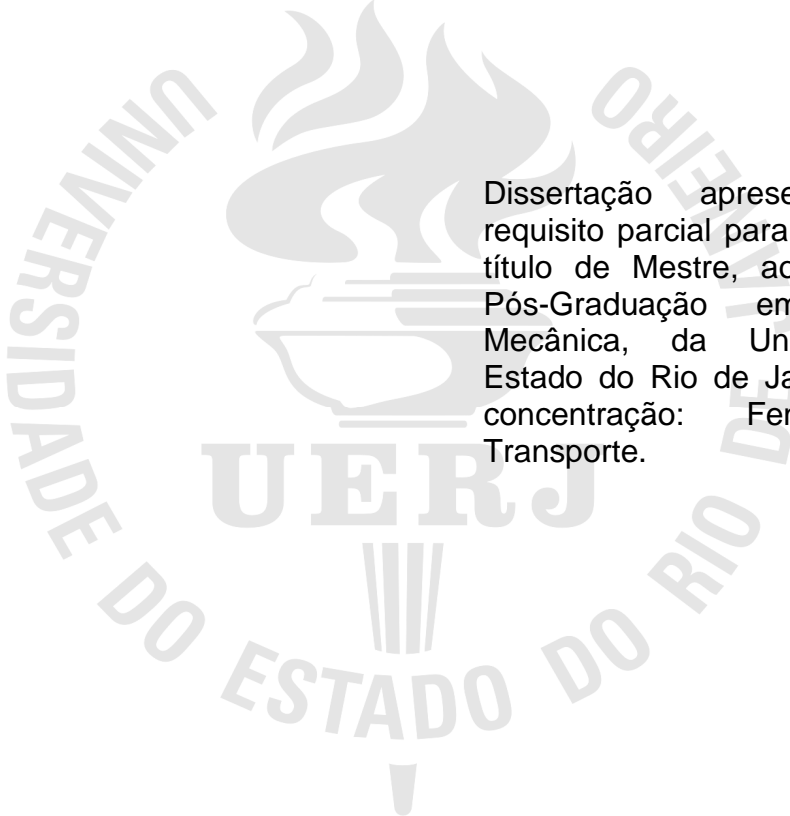
Faculdade de Engenharia

Nestor M. Solalinde

A FINITE ELEMENT BASED LEVEL SET APPROACH FOR THE SIMULATION OF TWO-PHASED FLOWS

Rio de Janeiro

2011

Nestor M. Solalinde

A finite element based level set approach for the simulation of two-phased flows

<div style="text-align: right">

Dissertação apresentada, como requisito parcial para a obtenção do título de Mestre, ao programa de Pós-Graduação em Engenharia Mecânica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Fenômenos de Transporte.

</div>

Orientador: Prof. D.Sc. Norberto Mangiavacchi

Rio de Janeiro

2011

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta tese, desde que citada a fonte.

_____   _____
             Assinatura   Data

Nestor M. Solalinde

A finite element based level set approach for the simulation of two-phased flows

Dissertação apresentada, como requisito parcial para a obtenção do título de Mestre, ao programa de Pós-Graduação em Engenharia Mecânica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Fenômenos de Transporte.

Aprovada em 31 de dezembro de 2011.

Banca examinadora:

_____

Prof. Dr. Norberto Mangiavacchi (Orientador)

Faculdade de Engenharia – UERJ

_____

Prof. Dr. Christian Emilio Schaerer Serra

Faculdad de Politécnica - UNA

_____

Prof. Dr. Fabricio Simeoni de Sousa

Instituto de Ciências Matemáticas e de Computação - USP

_____

Prof. Dr. Renato Nascimento Elias

COPPE - UFRJ

Rio de Janeiro

2011

DEDICATION

To my family.

ACKNOWLEDGMENTS

RESUMO

Neste trabalho, apresentamos uma abordagem level-set para simulação de fluidos bifasicos imiscíveis. O modelo consiste nas equações de Navier-Stokes incompressíveis, juntamente com uma equação de advecção para a função level-set. A abordagem CSF (Continuum Surface Force) é utilizada para modelar o efeito da tensão superficial. O método level-set é utilizado para capturar a interface entre os fluidos. Aqui, a função level-set é transportada e reinicializada em cada iteração. A discretização espacial é baseada em uma malha de elementos finitos não-estruturados que utiliza refinamento de malha adaptativa para maior precisão. Níveis de refinamento são calculados como altos na região próxima à interface e baixos longe dela. Para a discretização da velocidade e da pressão, espaços de elementos finitos padrão (LBB estável) são levados em consideração. A discretização da derivada no tempo resulta numa aproximação implícita. O algoritmo de elementos finitos é implementado usando a biblioteca de elementos finitos libmesh.

Palavras-chave: Simulação numérica; Método de elementos finitos; Fluidos multifásicos; Level-set; libMesh.

ABSTRACT

SOLALINDE, Nestor M. **A finite element based level set approach for the simulation of two-phased flows**. 2011. 71f. Dissertation (Master in Mechanical Engineering) – Faculty of Mechanical Engineering, State University of Rio de Janeiro, Rio de Janeiro, 2011.

      In this work, we present a level set approach for the simulation of two-phase immiscible newtonian fluids. The model consists on the incompressible Navier-Stokes equations, coupled with an advection equation for the level set function. The so-called Continuum Surface Force approach (CSF) is used to model the effect of surface tension. The level set method is used to capture the interface between the fluids. Here, the level set function is advected and reinitialized at each timestep. The spatial discretization is based on an unstructured Finite Element mesh that uses adaptive mesh refinement for higher accuracy. Refinement levels are set to be high on regions near the interface and low away from it. For the discretization of velocity and pressure fields, standard (LBB stable) finite element spaces are taken in consideration. Discretization of the time derivative is given by backward differentiation resulting in an implicit approximation. The finite element algorithm is implemented using the libmesh library.

Keywords: Numerical Simulation; Finite Element Method (FEM); Multiphase Flows; Level-Set method; libMesh.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER 1

INTRODUCTION

Many natural processes involve interactions between immiscible fluids. Bubbles and drops for example, play a major role in the interaction of the oceans with the atmosphere, and both, air bubbles near a free surface or cavitation bubbles are of major importance for the detection of submarines in naval applications. The study of bubble interaction is also important for the development of industrial equipments such as bubble-driven circulation systems used in metal processing operations such as steel making, ladle metallurgy, and the secondary refining of aluminum and copper (ESMAEELI; TRYGGVASON, 1998).

One of the main difficulties encountered when performing multiphase fluid simulations is related to the discontinuities at the fronts separating different fluids. A number of methods have been developed to approximate the fronts. Among these, the level set method, introduced by Osher and Sethian (OSHER; SETHIAN, 1988), has acquired popularity because of its algorithmic simplicity and its flexibility for handling topology changes. In this method, the fronts are represented by a zero level set of a function ϕ, that is advected by solving $\phi_t + \mathbf{u} \cdot \nabla \phi = \mathbf{0}$ where $\mathbf{u}$ is the velocity field. Most numerical procedures designed to solve this equation will introduce artificial diffusion leading to pronounced mass conservation errors. However, as pointed out by (SUSSMAN; SMEREKA; OSHER, 1994), mass conservation problems can be avoided by properly reinitializing the level set function.

In this work, the method is based on a level-set formulation discretized by a finite element technique. Surface tension forces acting at the interfaces separating the two fluids, as well as density and viscosity jumps across such interfaces have been integrated into the fi-

nite element framework. This method is based on the weak-formulation of the Navier Stokes Equations, where the singular surface tension forces, with the strength directly defined by the interface shape, are included through line integrals along the interfaces. The discontinuous density and viscosity are included in the finite element integrals.

As higher computational capabilities and resources become available day by day, adaptive mesh refining techniques are becoming popular and gaining interest from many researchers. These techniques are meant to reduce the computational costs of the algorithm and increase overall accuracy at the same time. As it is well known, the level set approach is commonly known to present mass conservation problems. By implementing adaptive mesh refinement, together with a mass conserving reinitialization function for the level set equation, we have observed that mass conservation problems of the level-set method are reduced to just a tiny fraction.

1.1 Objectives

The main objective of this work is to develop a two-phase flow simulation tool, that can be easily implemented and is capable of handling customized meshes and arbitrary boundary conditions, as well as topological changes such as merging and breaking of bubbles or drops.

Specific objectives are:

- Develop specific object oriented programming tools that provide flexibility and easy implementation of two-phased simulation models to new users.

- Describe the governing equations and the parameterizations inherent to two-phase flows.

- Describe the stabilized formulation and spatial discretization of both advection-diffusion and Navier-Stokes equations.

- Briefly describe the Object Oriented Programming (OOP) used in the numerical simulation.

- Present the computational results of benchmark validation models and their comparison with results available in literature.

1.2 Overview

This work is organized as follows:

Chapter 2 "*Mathematical Model*" gives an insight of the physical and mathematical grounds of the method used for the simulation of two-phased flows. This chapter introduces

the concepts behind simulation of incompressible fluids, surface tension and level set equations. At the end, a model summary is included to give the reader a general overview of the method.

Chapter 3 "*Time and Space Discretization*" exposes the weak, Galerkin and stabilized formulations as well as time discretization of: the convection-diffusion equation that is used to solve the advection and reinitialization of the level set equation, followed by, the Navier-Stokes equation for incompressible fluids. Furthermore, given the nonlinear nature of the Navier-Stokes equation, the linearization method used to solve the nonlinear problem is presented.

Chapter 4 "*Numerical Implementation*" explains how the set of principles and methods presented on previous chapters are implemented into a computational framework. It begins by presenting the general solution methodology. Then, the procedure for assembling the matrix and right hand side of the convection-diffusion and the Navier-Stokes equations is explained in detail. Furthermore, we describe the method used to define boundary conditions and to adaptively refine the mesh when needed. Finally, the solution algorithms are exposed in detail.

Chapter 5 "*Validation and Results*" presents the computational results of benchmark validation models. It begins by evaluating the Navier-Stokes solver accuracy, then, the pressure inside a static bubble and frequency of oscillation of an elliptical bubble and, finally, the rising velocity of a single bubble in a continuous phase.

Finally, Chapter 6 summarizes the work performed and discusses open issues which should be considered as future work.

CHAPTER 2

MATHEMATICAL MODEL

2.1 Introduction

In this chapter we will be setting the mathematical grounds for the simulation of multi-phasic flows. Particularly, our problem is reduced to the case of two immiscible fluids, where surface tension forces act at the interfaces separating both fluids with the intensity directly defined by the interface shape. The numerical method is based on a level-set formulation for incompressible multiphase flows. Here, the interface distance separating the two-fluids can be represented as the zero level set of a continuous function. In the level set method, the interface is embedded in a scalar distance field. This method was first introduced by (OSHER; SETHIAN, 1988).

The equations describing the immiscible multiphase incompressible flow are essentially the Navier Stokes equations for incompressible flow. The contribution of the surface tension forces $\mathbf{f}$ is, in addition to the gravity forces, added as a source term. Furthermore, we assume Dirichlet boundary conditions for the velocity and Newman for the pressure, with one Dirichlet point exception to remove the non-trivial null space of constant pressure solutions.

2.2 Governing Equations

Let us consider a domain $\Omega \in \Re^2$, that contains two different immiscible incompressible Newtonian phases (eg. fluid and gas). The time-dependent domains, which contains the phases, are denoted by $\Omega_1 = \Omega_1(t)$ and $\Omega_2 = \Omega_2(t)$ with $\overline{\Omega_1} \cup \overline{\Omega_2} = \overline{\Omega}$. The interface between the two phases ($\partial\Omega_1 \cap \partial\Omega_2$) is denoted by $\Gamma = \Gamma(t)$.

2.2.1   Mass Conservation

The mass conservation principle establish that the mass of a fluid volume (a volume that always contains the same fluid particles) is constant (HAUKE, 2008). Therefore we can write,

$$\frac{d}{dt}\mathbf{M} = \frac{d}{dt}\int_{\Omega_t} \rho dV = 0 \tag{2.1}$$

where Ω_t stands for a region of space occupied by the same fluid particles in a given time t. Now, we would like to express this in terms of a region in space R that is time independent and for that we apply the Reynolds transport theorem to (2.1) leading to,

$$\frac{d}{dt}\int_{\Omega_t} \rho dV = \int_R \frac{\partial \rho}{\partial t} dV + \oint_{\partial R} \rho(\mathbf{v} \cdot \mathbf{n}) dS = 0 \tag{2.2}$$

where $\mathbf{v}$ is the velocity field and $\mathbf{n}$ is the unit outward normal to R. Applying the divergence theorem to the second term on the right hand side we get,

$$\int_R \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) \right] dV = 0 \quad \Rightarrow \quad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{2.3}$$

Note that $\nabla \cdot (\rho \mathbf{v}) = \rho \nabla \cdot \mathbf{v} + \mathbf{v} \cdot \nabla \rho$, and we can rewrite equation (2.3) as

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} = 0 \tag{2.4}$$

where $\frac{D\rho}{Dt}$ is the material derivative of density.

We will see that the density parameter depends on the level set function ϕ that is not time independent, and is therefore not constant throw the whole domain. But again, taking in consideration that the density is variable only between one phase and the other, meaning, only in the interface region, and that there is no mass crossing the interface (the two phases are immiscible) we can neglect the material derivative term of density in equation (2.4). That leaves us with

$$\rho \nabla \cdot \mathbf{v} = 0 \quad \text{or} \quad \nabla \cdot \mathbf{v} = 0 \tag{2.5}$$

2.2.2   Conservation of Momentum

Euler's first law of conservation of the linear momentum states that in an inertial frame the time rate of change of linear momentum of an arbitrary portion of a continuous body is equal to the total applied force acting on the considered portion (HAUKE, 2008), and is expressed as,

$$\frac{d}{dt}\int_{\Omega_t}\rho\mathbf{v}dV = F_S + F_V = \oint_{\partial\Omega}\sigma\cdot\mathbf{n}dS + \int_{\Omega_t}\mathbf{b}dV \tag{2.6}$$

where F_S and F_V are the surface forces and body forces respectively, where σ is the Cauchy stress tensor and $\mathbf{b}$ is body force per volume unit. When the active body force is only due to the gravitational field we can write $\mathbf{b} = \rho\mathbf{g}$, where $\mathbf{g}$ is the gravitational acceleration. Applying the Reynolds transport theorem to the left hand side of equation (2.6) we obtain,

$$\int_{\Omega}\frac{\partial(\rho\mathbf{v})}{\partial t}dV + \oint_{\partial\Omega}(\rho\mathbf{v}\otimes\mathbf{v})\cdot\mathbf{n}dS = \oint_{\partial\Omega}\sigma\cdot\mathbf{n}dS + \int_{\Omega}\mathbf{b}dV \tag{2.7}$$

Applying the divergence theorem,

$$\int_{\Omega}\frac{\partial(\rho\mathbf{v})}{\partial t}dV + \int_{\Omega}\nabla\cdot(\rho\mathbf{v}\otimes\mathbf{v})dV = \int_{\Omega}\nabla\cdot\sigma dV + \int_{\Omega}\mathbf{b}dV \tag{2.8}$$

Recalling from section (2.2.1) that $\mathbf{v}\frac{\partial\rho}{\partial t} + \mathbf{v}\nabla\cdot(\rho\mathbf{v}) = 0$, we can write the left hand side of equation (2.8) as,

$$\begin{aligned}\frac{\partial(\rho\mathbf{v})}{\partial t} + \nabla\cdot(\rho\mathbf{v}\otimes\mathbf{v}) &= \frac{\partial(\rho\mathbf{v})}{\partial t} + \mathbf{v}\nabla\cdot(\rho\mathbf{v}) + \rho\nabla\mathbf{v}\cdot\mathbf{v}\\ &= \rho\frac{\partial\mathbf{v}}{\partial t} + \mathbf{v}\frac{\partial\rho}{\partial t} + \mathbf{v}\nabla\cdot(\rho\mathbf{v}) + \rho\mathbf{v}\cdot\nabla\mathbf{v}\\ &= \rho\frac{\partial\mathbf{v}}{\partial t} + \rho\mathbf{v}\cdot\nabla\mathbf{v}\end{aligned}$$

Subsequently, we can rewrite equation (2.8) as,

$$\rho\frac{\partial\mathbf{v}}{\partial t} + \rho\mathbf{v}\cdot\nabla\mathbf{v} = \nabla\cdot\sigma + \mathbf{b} \tag{2.9}$$

2.2.3 Navier Stokes Equations

Let us now consider a flow region $\Omega\in\Re^{n_{sd}}$, where $n_{sd} = 2$ or 3. The domain Ω occupied by the fluid will be assumed bounded (finite size). The boundary $\partial\Omega$ of the fluid domain is assumed to be Lipschitz continuous, meaning that it is a closed and sufficient regular surface. Then, the time-dependent flow of a viscous incompressible fluid is governed by the following form of the momentum equation (2.9) and the mass-conservation equation (2.4), called the Navier-Stokes equations:

$$\rho\frac{\partial\mathbf{v}}{\partial t} + \rho\mathbf{v}\cdot\nabla\mathbf{v} = \nabla\cdot\sigma + \mathbf{b} \qquad \text{in }\Omega \tag{2.10a}$$

$$\nabla\cdot\mathbf{v} = 0 \qquad \text{in }\Omega \tag{2.10b}$$

2.2.4   Newtonian fluids

A common practice is to decompose the Cauchy stress tensor σ (or σ_{ij} written in index notation), into the sum of an isotropic part $-p\delta_{ij}$ and a remaining non-isotropic part τ_{ij}, the deviatoric stress tensor:

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij} \tag{2.11}$$

Following (DONEA; HUERTA, 2003), in a Newtonian fluid, it is assumed that the stress tensor and the strain rate tensor are linearly related. The stress-strain rate relationship is given by

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij} = -p\delta_{ij} + \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + \lambda\frac{\partial v_k}{\partial x_k}\delta_{ij} \tag{2.12}$$

where μ is the fluid dynamic viscosity and λ the so-called second coefficient of viscosity. For an incompressible fluid one has $\nabla \cdot \mathbf{v} = 0$ and consequently the above relation reduces to the Stokes' law

$$\sigma_{ij} = -p\delta_{ij} + \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) \tag{2.13}$$

that in compact form reads

$$\sigma = -p\mathbf{I} + 2\mu\varepsilon(\mathbf{u}) \tag{2.14}$$

here, ε is the deformation rate tensor,

$$\varepsilon(\mathbf{u}) = \frac{1}{2}[\nabla\mathbf{u} + (\nabla\mathbf{u})^T] \tag{2.15}$$

2.2.5   Surface Tension

The effect of the surface tension can be expressed in terms of a localized force at the interface, by the so-called continuum surface force (CSF) model.(BRACKBILL; KOTHE; ZEMACH, 1992)

$$\mathbf{f} = \sigma k\nabla H_\Gamma \tag{2.16}$$

where σ is the surface tension coefficient, k is the curvature and ∇H_Γ is the Heaviside function with support on Γ, this is, $H_\Gamma = H(\phi)$. The mean curvature is computed from the level set function as shown on section 2.4.3. Note that we can also express equation (2.16) in terms of the derivative of the Heaviside function as follows,

$$\mathbf{f} = \sigma k\delta_\Gamma\mathbf{n} \tag{2.17}$$

where δ_Γ is the derivative of the Heaviside function and $\mathbf{n}$ is the unit vector normal to the interface.

2.3   Nondimensionalization

With the purpose of parameterize our problem, we will write the Navier-Stokes equations in nondimensional form. Now let us rewrite the navier stokes equations (2.10) using equations (2.14), (2.15) and (2.16), therefore assuming that we are working with newtonian fluids, a gravitational force field $\mathbf{g}$ and surface tension forces acting on the interface Γ

$$\rho\frac{\partial \mathbf{v}}{\partial t} + \rho\mathbf{v}\cdot\nabla\mathbf{v} = -\nabla\cdot p + \nabla\cdot[\mu(\nabla\mathbf{v}+\nabla\mathbf{v}^T)] + \rho\mathbf{g} + \sigma k\delta_\Gamma\mathbf{n} \qquad \text{in } \Omega \qquad (2.18a)$$

$$\nabla\cdot\mathbf{v} = 0 \qquad\qquad\qquad \text{in } \Omega \qquad (2.18b)$$

and let us define the following nondimensionalization parameters,

$$\rho = \rho_\infty\rho^* \qquad\qquad x = Lx^* \qquad\qquad \mathbf{g} = g_\infty\mathbf{g}^* \qquad k = \frac{1}{L}k^*$$

$$\mu = \mu_\infty\mu^* \qquad\qquad \mathbf{v} = U\mathbf{v}^* \qquad\qquad t = \frac{L}{U}t^* \qquad \delta_\Gamma = \frac{1}{L}\delta_\Gamma^*$$

$$p = \rho_\infty U^2 p^* \qquad \frac{\partial}{\partial t} = \frac{U}{L}\frac{\partial}{\partial t^*} \qquad \nabla = \frac{1}{L}\nabla^*$$

where the asterisk indicates a nondimensional variable. Substituting these parameters on equations (2.10a) and (2.10b) we get,

$$\rho^*\left(\frac{\rho_\infty U^2}{L}\right)\frac{\partial \mathbf{v}^*}{\partial t^*} + \rho^*\left(\frac{\rho_\infty U^2}{L}\right)\mathbf{v}^*\cdot\nabla^*\mathbf{v}^* = -\left(\frac{\rho_\infty U^2}{L}\right)\nabla^*\cdot p^*$$

$$+\frac{\mu_\infty U}{L^2}\nabla^*\cdot[\mu^*(\nabla^*\mathbf{v}^*+\nabla^*\mathbf{v}^{*T})] + \rho_\infty g_\infty\rho^*\mathbf{g}^* + \frac{\sigma}{L^2}k^*\delta_\Gamma^*\mathbf{n} \qquad (2.19a)$$

$$\nabla^*\cdot\mathbf{v}^* = 0 \qquad\qquad\qquad\qquad (2.19b)$$

Note however that σ is not dimensionless. If we multiply (2.19a) by $L/\rho_\infty U^2$ we get,

$$\rho^*\frac{\partial \mathbf{v}^*}{\partial t^*} + \rho^*\mathbf{v}^*\cdot\nabla^*\mathbf{v}^* = -\nabla^*\cdot p^* + \left(\frac{\mu_\infty}{\rho_\infty LU}\right)\nabla^*\cdot[\mu^*(\nabla^*\mathbf{v}^*+\nabla^*\mathbf{v}^{*T})]$$

$$+\left(\frac{Lg_\infty}{U^2}\right)\rho^*\mathbf{g}^* + \left(\frac{\sigma}{\rho_\infty U^2 L}\right)k^*\delta_\Gamma^*\mathbf{n} \qquad (2.20a)$$

$$\nabla^*\cdot\mathbf{v}^* = 0 \qquad\qquad\qquad\qquad (2.20b)$$

And equation (2.20) can be written as,

$$\rho^*\frac{\partial \mathbf{v}^*}{\partial t^*} + \rho^*\mathbf{v}^*\cdot\nabla^*\mathbf{v}^* = -\nabla^*\cdot p^* + \frac{1}{Re}\nabla\cdot[\mu^*(\nabla^*\mathbf{v}^*+\nabla^*\mathbf{v}^{*T})]$$

$$+\frac{1}{Fr^2}\rho^*\mathbf{g}^* + \frac{1}{We}k^*\delta_\Gamma^*\mathbf{n} \tag{2.21a}$$

$$\nabla^* \cdot \mathbf{v}^* = 0 \tag{2.21b}$$

Where Re, Fr and We are the nondimensional Reynolds, Froude and Weber numbers respectively. These are depicted as follows(HAUKE, 2008),

Reynolds Number. The Reynolds number steams from dividing the convective term by the viscous diffusion term, representing the ratio of inertial to viscous forces, and is given by,

$$Re = \frac{\rho_\infty L U}{\mu_\infty} \tag{2.22}$$

Froude Number. Similarly to the Reynolds number, the Froude number is the ration of inertial forces to body forces, and is given by,

$$Fr = \frac{U}{\sqrt{g_\infty L}} \tag{2.23}$$

Weber Number. The Weber number is a dimensionless number that controls the significance of the surface tension force. It is given by,

$$We = \frac{\rho_\infty U^2 L}{\sigma} \tag{2.24}$$

When studying the dynamics of bubbles or drops, the Morton (Mo) and Eotvos(Eo) nondimensional numbers can also be useful. These are used to caracterize the shape of bubbles or drops. They are defined as,

$$Mo \quad = \frac{g_\infty \mu^4}{\rho_\infty \sigma^3} \tag{2.25}$$

$$Eo \quad = \frac{\Delta \rho g L^2}{\sigma} \tag{2.26}$$

The Morton and Eotvos numbers can also be expressed in terms of the Reynolds, Froude and Weber numbers through the following equations,

$$Eo \quad = \frac{We}{Fr^2} \tag{2.27}$$

$$Mo \quad = \frac{We^3}{Fr^2 Re^4} \tag{2.28}$$

2.4   Level-Set Equations

Originally introduced by (OSHER; SETHIAN, 1988), the level-set method is a simple and versatile method for computing and analyzing the motion of an interface Γ in two or three

dimensions. It allow us to compute and analyze the subsequent motion of the interface under a velocity field. The interface is captured as the zero level set of a smooth (Lipschitz continuous) distance function ϕ, positive inside Ω_1, negative outside Ω_1 and zero on Γ. Topological merging and breaking are well defined and easily performed.

The jumps in the coefficients, μ and ρ can be described using the level set function (which has its zero level precisely at the interface Γ) in combination with the Heaviside function,

$$H(\zeta) = 0 \text{ for } \zeta < 0, \qquad H(\zeta) = 1 \text{ for } \zeta > 0 \tag{2.29}$$

$$\rho(\phi) = \rho_1 + (\rho_2 - \rho_1)H(\phi) \tag{2.30}$$

$$\mu(\phi) = \mu_1 + (\mu_2 - \mu_1)H(\phi) \tag{2.31}$$

where we can take $H(0) = 1/2$. However, as a result of the discontinuous density and viscosity jumps, numerical instabilities are presented, and since $\delta_\Gamma = 0$ almost everywhere except on the interface (which has measure 0), it seems unlikely that any standard numerical approximation based on sampling will give a good approximation of the integral of equation (2.16). Thus, we define the smeared-out Heaviside function (OSHER; FEDKIW, 2003),

$$H_\epsilon(\phi) = \left\{ \begin{array}{cc} 0 & \phi < -\epsilon \\ \frac{1}{2} + \frac{\phi}{2\epsilon} + \frac{1}{2\pi}sin\left(\frac{\pi\phi}{\epsilon}\right) & -\epsilon \leq \phi \leq \epsilon \\ 1 & \epsilon < \phi \end{array} \right\} \tag{2.32}$$

where ϵ is a parameter that determines the size of the bandwidth of numerical smearing. A typically good value is $\epsilon = 1.5\Delta x$ (OSHER; FEDKIW, 2003), where Δx can be taken as an approximate measure of the grid. Furthermore, we can replace equation (2.16) with

$$\mathbf{f} = \sigma k \nabla H_\epsilon \tag{2.33}$$

Additionally, we define the smeared-out delta function as,

$$\delta_\epsilon = \frac{\partial H_\epsilon}{\partial \phi} = \left\{ \begin{array}{cc} 0 & \phi < -\epsilon \\ \frac{1}{2\epsilon} + \frac{1}{2\epsilon}cos\left(\frac{\pi\phi}{\epsilon}\right) & -\epsilon \leq \phi \leq \epsilon \\ 0 & \epsilon < \phi \end{array} \right\} \tag{2.34}$$

2.4.1 Level-Set Advection

The level set function is initialized as a signed distance function, carrying information about the closest distance to any interface separating the two fluids. To determine the evolution of the interfaces, the level set function is advected over the domain by the velocity field. The

advection equation is given by the following equation,

$$\phi_t + \mathbf{u} \cdot \nabla \phi = \mathbf{0} \qquad (2.35)$$

which is equivalent to a linear convection-diffusion problem, where the diffusion term is equal to zero.

The strong form of the problem is given as follows: Given a scalar $k \in R$ and a vector field $\mathbf{b}$ defined on a region $\Omega \subset R^d$, find $\phi(x, t)$ such that:

$$
\begin{aligned}
\dot{\phi} - k\Delta\phi + \mathbf{b} \cdot \nabla\phi &= f & on & \quad \Omega \times I \\
\phi &= g & on & \quad \Gamma \times I \\
\phi(x, 0) &= \phi^0 & on & \quad \Omega
\end{aligned}
\qquad (2.36)
$$

with $f = 0$, $\boldsymbol{b} = \boldsymbol{u}$, and $k = 0$.

Dado um escalar $k \in R$ e um campo vetorial $\mathbf{b}$ definido na região $\Omega \subset R^d$, encontre $\phi(x, t)$ tal que:

$$
\begin{aligned}
\dot{\phi} - k\Delta\phi + \mathbf{b} \cdot \nabla\phi &= f & on & \quad \Omega \times I \\
\phi &= g & on & \quad \Gamma \times I \\
\phi(x, 0) &= \phi^0 & on & \quad \Omega
\end{aligned}
\qquad (2.37)
$$

onde $f = 0$, $\boldsymbol{b} = \boldsymbol{u}$, e $k = 0$.

2.4.2   Level-Set Reinitialization

It has been shown (SUSSMAN; SMEREKA; OSHER, 1994) that it is critical that the level set function remains a distance function in regions close to the interface, in order to obtain acceptable accuracy in the computation of the unit normal and the mean curvature. After advection, the level set function does not necessarily correspond to a distance function any more. Keeping ϕ a distance function is achieved by reinitialization of the level set function, given by the following equation introduced by (SUSSMAN; SMEREKA; OSHER, 1994),

$$\frac{\partial\varphi}{\partial\tau} = S(\varphi_0)(1 - |\nabla\varphi|) \qquad (2.38)$$

where φ_0 is the solution of equation (2.37) for a given time, $\varphi(\tau)$ is the reinitialized level set function at artificial time τ and $S(\varphi)$ is a sign function, given by

$$
S(\varphi) = \left\{
\begin{array}{cl}
-1 & \text{for } \varphi < 0 \\
0 & \text{for } \varphi = 0 \\
1 & \text{for } \varphi > 0
\end{array}
\right\}
\qquad (2.39)
$$

Numerical tests, indicate that better results are obtained when $S(\varphi_0)$ is numerically smeared out, so a better choice is determined by (PENG et al., 1999) as

$$S(\varphi, h) = \frac{\varphi}{\sqrt{\varphi^2 + |\nabla\varphi|^2 (h)^2}} \tag{2.40}$$

which has shown to yield improved and stable results, especially when the initial φ_0 is a poor estimate of signed distance (OSHER; FEDKIW, 2003).

By solving this equation, we obtain a distance function with the same zero level set of φ_0. Equation (2.38) can also be written as: (TORNBERG; ENGQUIST, 2000)

$$\frac{\partial\varphi}{\partial\tau} + w \cdot \nabla\varphi = S(\varphi_0), \qquad w = S(\varphi_0)\frac{\nabla\varphi}{|\nabla\varphi|} \tag{2.41}$$

which is a purely convective system, with the velocity field defined by w and the force function f defined by $f = S(\varphi)$. This equation can be solved as a convection diffusion equation as described in section 3.2.

A method presented by (SUSSMAN; FATEMI, 1999) suggests an improvement to the standard reinitialization procedure. The method states that if the interface does not move during reinitialization, the area is preserved. On the other hand, one can preserve the area while allowing the interface to move, implying that the proposed constraint is weaker than it should be. The local constraint is implemented by the addition of a correction term to the right hand side of equation (2.38),

$$\frac{\partial\varphi}{\partial\tau} = S(\varphi_0)(1 - |\nabla\varphi|) + \lambda\delta_\epsilon(\varphi)|\nabla\varphi| \tag{2.42}$$

We can rewrite this as follows,

$$\frac{\partial\varphi}{\partial\tau} = \frac{\partial\widetilde{\varphi}}{\partial\tau} + \lambda\delta_\epsilon(\widetilde{\varphi})|\nabla\widetilde{\varphi}| \quad \text{with,} \tag{2.43}$$

$$\frac{\partial\widetilde{\varphi}}{\partial\tau} = S(\varphi_0)(1 - |\nabla\widetilde{\varphi}|) \tag{2.44}$$

The mentioned constrain is defined by

$$\int_{\Omega_e} H_\epsilon(\varphi)\,d\Omega = 0 \tag{2.45}$$

where Ω_e is an individual cell and $H_\epsilon(\phi)$ is the smeared-out heaviside function defined by equation (2.59). This is equivalent to

$$\int_{\Omega_e} \dot{H}_\epsilon(\varphi)\,\varphi_t\,d\Omega = \int_{\Omega_e} \delta_\epsilon(\varphi)(\widetilde{\varphi}_t + \lambda\delta_\epsilon(\varphi)|\nabla\varphi|)d\Omega = 0 \tag{2.46}$$

This way, a particular $\lambda_{i,j}$ is determined on each cell by the following equation,

$$\lambda_e = -\frac{\int_{\Omega_e} \delta_\epsilon(\varphi) \left(\frac{\widetilde{\varphi}^{n+1} - \widetilde{\varphi}^n}{\Delta t}\right) d\Omega}{\int_{\Omega_e} \delta_\epsilon{}^2(\varphi) |\nabla\varphi| \, d\Omega} \qquad (2.47)$$

where equation (2.43) is used to compute $\widetilde{\varphi}^{n+1}$ from $\widetilde{\varphi}^n$.

In summary, the initial guess $\widetilde{\varphi}^{n+1}$ obtained from equation (2.43) is replaced with a corrected $\widetilde{\varphi}^{n+1} + \Delta t\,\lambda\delta_\epsilon(\varphi)\,|\nabla\varphi|$, where λ is calculated on each cell using equation (2.47).

2.4.3   Mean Curvature

Whenever the level set function remains a perfect distance function, i.e. $|\nabla\phi| = 1$, without imperfections, we can calculate the unit normal and the mean curvature using the following equations,

$$\mathbf{n} \;=\; \nabla\phi \qquad (2.48a)$$

$$k \;=\; \nabla\cdot\mathbf{n} \qquad (2.48b)$$

However, if the level set function is not a perfect distance function or if it presents even the smallest imperfections, then these will be significantly magnified by the derivatives and even more by second derivatives. To reduce these errors we make use of a new methodology in which the unit normal is corrected using a reinitialization procedure (given that $\nabla\phi\cdot\nabla\frac{\partial\phi}{\partial x_i} = 0$) and the calculation of the mean curvature is made by using directional derivatives. The new equations are then presented as follows (2-dimensional case),

$$nx + \Delta T(S(\phi)\nabla\phi)\cdot\nabla nx = \frac{\partial\phi}{\partial x} \qquad (2.49a)$$

$$ny + \Delta T(S(\phi)\nabla\phi)\cdot\nabla ny = \frac{\partial\phi}{\partial y} \qquad (2.49b)$$

$$\mathbf{n} = \frac{(nx, ny)}{\|(nx, ny)\|} \qquad (2.49c)$$

$$\mathbf{n}_\perp = (-ny, nx) \text{ (see that } \mathbf{n}\cdot\mathbf{n}_\perp = 0) \qquad (2.49d)$$

$$k = -ny(\mathbf{n}_\perp\cdot\nabla nx) + nx(\mathbf{n}_\perp\cdot\nabla ny) \qquad (2.49e)$$

where ΔT is a constant that can generally be set to be numerically equal to the interface jump smearing constant ϵ. The 3-dimensional case can be stated as follows,

$$nx + \Delta T(S(\phi)\nabla\phi)\cdot\nabla nx = \frac{\partial\phi}{\partial x} \qquad (2.50a)$$

$$ny + \Delta T(S(\phi)\nabla\phi) \cdot \nabla ny = \frac{\partial \phi}{\partial y} \tag{2.50b}$$

$$nz + \Delta T(S(\phi)\nabla\phi) \cdot \nabla nz = \frac{\partial \phi}{\partial z} \tag{2.50c}$$

$$\mathbf{n} = \frac{(nx, ny, nz)}{\|(nx, ny, nz)\|} \tag{2.50d}$$

$$\mathbf{n_{\perp 1}} = (nx_{\perp 1}, ny_{\perp 1}, nz_{\perp 1}) = \frac{(-ny, nx, 0)}{\|(-ny, nx, 0)\|} \tag{2.50e}$$

$$\mathbf{n_{\perp 2}} = (nx_{\perp 2}, ny_{\perp 2}, nz_{\perp 2}) = \mathbf{n} \times \mathbf{n_{\perp 1}} \tag{2.50f}$$

$$k = nx_{\perp 1}(\mathbf{n_{\perp 1}} \cdot \nabla nx) + ny_{\perp 1}(\mathbf{n_{\perp 1}} \cdot \nabla ny) + nz_{\perp 1}(\mathbf{n_{\perp 1}} \cdot \nabla nz)$$
$$+ nx_{\perp 2}(\mathbf{n_{\perp 2}} \cdot \nabla nx) + ny_{\perp 2}(\mathbf{n_{\perp 2}} \cdot \nabla ny) + nz_{\perp 2}(\mathbf{n_{\perp 1}} \cdot \nabla nz) \tag{2.50g}$$

where $\times$ is the cross-product operator. Note that $\mathbf{n_{\perp 1}} \cdot \mathbf{n} = 0$, $\mathbf{n_{\perp 2}} \cdot \mathbf{n} = 0$ and $\mathbf{n_{\perp 1}} \cdot \mathbf{n_{\perp 2}} = 0$. Note also that an exception must be made for the case where $\mathbf{n} = (0, 0, k)$.
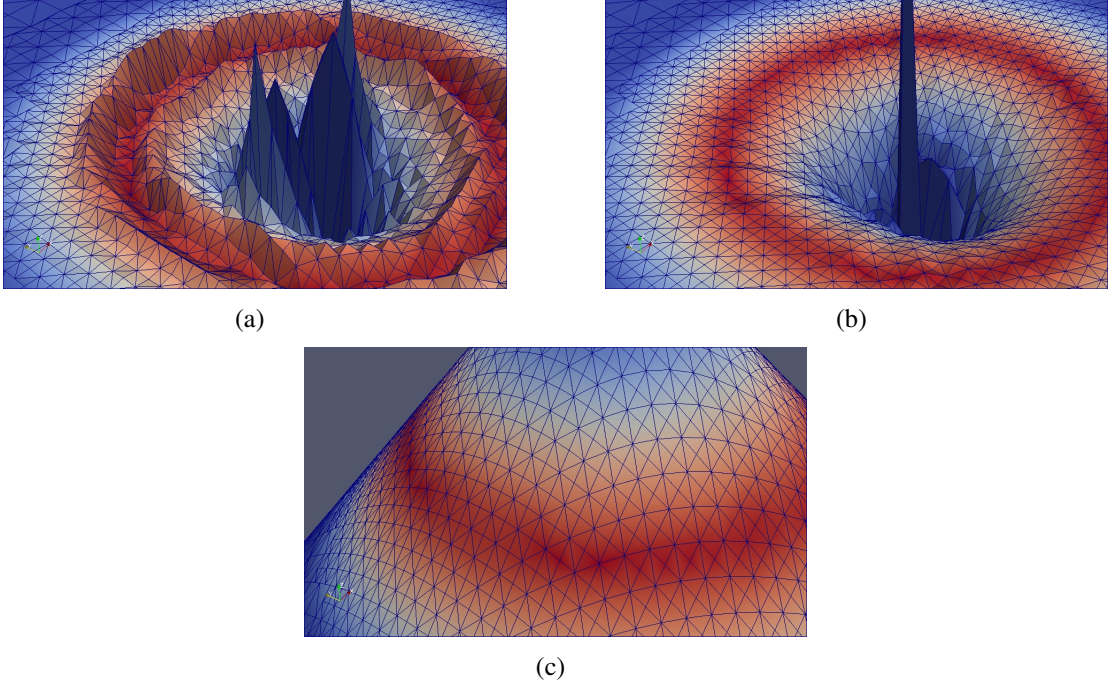
Figure 2.1: Curvature calculation: (a) using equation set (2.48), (b) using equation set (2.49), (c) level set function for (a) and (b).

2.5 Model Summary

Let us consider a domain $\Omega \in \Re^2$, that contains two different immiscible incompressible Newtonian phases (eg. fluid and gas). The time-dependent domains which contain the phases are denoted by $\Omega_1 = \Omega_1(t)$ and $\Omega_2 = \Omega_2(t)$ with $\overline{\Omega_1} \cup \overline{\Omega_2} = \overline{\Omega}$. The interface between the two phases $(\partial\Omega_1 \cap \partial\Omega_2)$ is denoted by $\Gamma = \Gamma(t)$. The equations describing the immiscible

multiphase incompressible flow are essentially the Navier Stokes equations for incompressible flow. The contribution of the surface tension forces $\mathbf{f}$ is, in addition to the gravity forces, added as a source term. The equations, assuming continuity of the velocity across the interface, can be written:

$$\rho_r(\phi)\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) = -\nabla \cdot p + \frac{\rho_r(\phi)}{Fr^2}\mathbf{g} + \frac{1}{We}\mathbf{f} \tag{2.51}$$

$$+ \frac{1}{Re}\nabla \cdot [\mu_r(\phi)(\nabla \mathbf{u} + \nabla \mathbf{u^T})]$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2.52}$$

$$\phi_t + \mathbf{u} \cdot \nabla \phi = 0 \tag{2.53}$$

where $\mathbf{u}$ is the velocity field, p is the pressure field, μ_r and ρ_r are the discontinuous relative viscosity and density, $\mathbf{g}$ represents the gravitational acceleration field and ϕ is the level set equation. In equation (2.51), Re, Fr and We are the non-dimensional Reynolds, Froude and Weber numbers. Furthermore, we assume Dirichlet boundary conditions for the velocity and Newman for the pressure, with one Dirichlet point exception to remove the non-trivial null space of constant pressure solutions.

The effect of the surface tension can be expressed in terms of a localized force at the interface, by the so-called continuum surface force (CSF) model.(BRACKBILL; KOTHE; ZEMACH, 1992)

$$\mathbf{f} = \sigma_r k \nabla H_\Gamma \tag{2.54}$$

where σ_r is the relative surface tension coefficient, k is the curvature and H_Γ is the Heaviside function with support on Γ. The unit normal and the mean curvature are computed using the following equations,

$$nx + \Delta T(S(\phi)\nabla \phi) \cdot \nabla nx = \frac{\partial \phi}{\partial x} \tag{2.55a}$$

$$ny + \Delta T(S(\phi)\nabla \phi) \cdot \nabla ny = \frac{\partial \phi}{\partial y} \tag{2.55b}$$

$$\mathbf{n} = \frac{(nx, ny)}{\|(nx, ny\|} \tag{2.55c}$$

$$\mathbf{n}_\perp = (-ny, nx) \tag{2.55d}$$

$$k = -ny(\mathbf{n}_\perp \cdot \nabla nx) + nx(\mathbf{n}_\perp \cdot \nabla ny) \tag{2.55e}$$

The jumps in the coefficients, μ and ρ can be described using the level set function

(which has its zero level precisely at the interface Γ) in combination with the Heaviside function,

$$H(\zeta) = 0 \text{ for } \zeta < 0, \qquad H(\zeta) = 1 \text{ for } \zeta > 0 \tag{2.56}$$

$$\rho(\phi) = \rho_1 + (\rho_2 - \rho_1)H(\phi) \tag{2.57}$$

$$\mu(\phi) = \mu_1 + (\mu_2 - \mu_1)H(\phi) \tag{2.58}$$

As a result of the discontinuous density and viscosity jumps, numerical instabilities are presented, and since $\delta_\Gamma = 0$ almost everywhere except on the interface (which has measure 0), it seems unlikely that any standard numerical approximation based on sampling will give a good approximation of the integral of equation (2.54). Thus, we define the smeared-out Heaviside function,

$$H_\epsilon\left(\phi\right) = \left\{ \begin{array}{cc} 0 & \phi < -\epsilon \\ \frac{1}{2} + \frac{\phi}{2\epsilon} + \frac{1}{2\pi} sin\left(\frac{\pi\phi}{\epsilon}\right) & -\epsilon \leq \phi \leq \epsilon \\ 1 & \epsilon < \phi \end{array} \right\} \tag{2.59}$$

where ϵ is a parameter that determines the size of the bandwidth of numerical smearing. Then, the smeared-out delta function is defined as,

$$\delta_\epsilon = \frac{\partial \mathrm{H}_\epsilon}{\partial \phi} = \left\{ \begin{array}{cc} 0 & \phi < -\epsilon \\ \frac{1}{2\epsilon} + \frac{1}{2\epsilon} \cos\left(\frac{\pi}{\epsilon}\right) & -\epsilon \leq \phi \leq \epsilon \\ 0 & \epsilon < \phi \end{array} \right\} \tag{2.60}$$

where ϵ is determined as above, and we can replace equation (2.54) with

$$\mathbf{f} = \sigma_r k \delta_\epsilon \mathbf{n} \tag{2.61}$$

To complete the general formulation given by equations (2.51), (2.52) and (2.53), we must now include the condition of keeping the level set function a signed distance function. We do this by reinitializing the level set function with the following equation introduced by Sussman et. al. (SUSSMAN; SMEREKA; OSHER, 1994)

$$\frac{\partial \varphi}{\partial \tau} = S\left(\varphi_0\right)\left(1 - |\nabla\varphi|\right) \tag{2.62}$$

where φ_0 is the solution of equation (2.53) for a given time, $\varphi\left(\tau\right)$ is the reinitialized level set function at artificial time τ and $S(\varphi)$ is a sign function.

TIME AND SPACE DISCRETIZATION

This work makes use of the finite element method for the spatial discretization. Among the basic advantages of the method which have led to its widespread acceptance and use are the ease in modeling complex geometries, the consistent treatment of differential-type boundary conditions and the possibility to be programmed in a flexible and general purpose format.

We will start this chapter by setting up mathematical preliminar definitions to be used later on. Subsequently, we will expose the weak, Galerkin, stabilized formulation and time discretization scheme for the Convection-Diffusion equation which is the general formulation for both the level-set equation (2.37) and the reinitialization equation (2.41). Furthermore, we will proceed likewise to the discrete stabilized form of the Navier-Stokes equation (2.21). Finally, given the nonlinear nature of the Navier-Stokes equation, we will present the linearization method used to solve the nonlinear problem.

3.1 Preliminary definitions

Consider a spatial region (or domain) $\Omega \subset \mathbb{R}^{n_{sd}}$ discretized with piecewise smooth boundary Γ. Here, $n_{sd} = 2$ or 3 denotes the number of spatial dimensions. Following (DONEA; HUERTA, 2003), we shall denote by $\mathcal{L}_2(\Omega)$ the space of functions that are square integrable over the domain Ω. This space is equipped with the standard inner product,

$$(u, v) = \int_{\Omega} uv d\Omega \qquad \text{and norm} \qquad \|v\|_0 = (v, v)^{1/2}$$

Furthermore, for any non-negative integer k, we define the Sobolev space $H^k(\Omega)$: given $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{n_{sd}}) \in \mathbb{N}^{n_{sd}}$ and the non-negative integer $|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_{n_{sd}}$,

$$H^k(\Omega) = \left\{ u \in \mathcal{L}_2(\Omega) \left| \frac{\partial^{|\alpha|} u}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_{n_{sd}}^{\alpha_{n_{sd}}}} \in \mathcal{L}_2(\Omega) \forall |\alpha| \leq k \right. \right\}$$

Note that the Sobolev space for $k = 1$ is defined by

$$H^1(\Omega) = \left\{ v \in \mathcal{L}_2(\Omega) \left\| \frac{\partial v}{\partial x_i} \in \mathcal{L}_2(\Omega), \forall i \in \mathbb{N}, i \leq n_{sd} \right. \right\}$$

We shall also use frequently the subspace

$$H_0^1(\Omega) = \{ v \in H^1(\Omega) | v = 0 \text{ on } \Gamma \}$$

For the particular case of vector functions belonging to $\left[H^k(\Omega) \right]^m$, the space of vector functions with m components, the inner product is given by

$$(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{u} \cdot \mathbf{v} d\Omega \tag{3.1}$$

In addition, inner products for functions defined along boundaries are denoted by

$$(u, h)_{\Gamma} = \int_{\Gamma} u h d\Gamma \tag{3.2}$$

Note that there is no ambiguity in using the same notation to represent the inner product of vector valued functions or scalars.

Recalling equation (2.14) and (2.15) of section 2.2.4, note that neither of these equations are in nondimensional form, for which it is convenient to define

$$\sigma^* = -p^* \mathbf{I} + \frac{2\mu_r}{Re} \varepsilon(\mathbf{u}^*) \tag{3.3}$$

where σ^*, p^* and $\mathbf{u}^*$ are nondimensionalized variables.

3.2 Convection-Diffusion equation

As mentioned before, the solution of the convection-diffusion equation plays a mayor role as it represents the general case for the solution of the level-set advection and reinitialization.

Level Set advection.   The Level-Set advection equation is a convection diffusion equation where particularly $f = 0$, $\Gamma = \Gamma_N$, $h = 0$, $k \approx 0$ and the velocity $\mathbf{b}$ is set equal to the velocity $\mathbf{u}$ given by the navier stokes equation. Ideally, we should be able to solve this system setting the diffusion term $k = 0$ but due to the unstable nature of the numerical problem it is a common practice to use non-zero values. However, since we are applying streamline diffusion we are able to use a fairly low magnitudes of k. Since we are looking for accuracy on the solution of the level-set advection equation, we'll use $\theta = 0.5$ (centered differences), therefore obtaining second order accuracy for the time discretization.

Reinitialization equation.   The reinitialization equation (2.41) is a purely convective system, with the velocity field defined by $\mathbf{w}$ and the force function f defined by $f = S(\varphi)$. Therefore, we can apply the same procedure used for the advection of the level set equation, just that for this case, we will use an implicit time discretization scheme, because we are looking for fast, robust convergence and only the steady state solution is of interest. Same as with the advection of the level set equation, we are forced to include small amounts of artificial diffusion to ensure stability. However, mass losses still remain insignificant due to the use of the mass conserving technique described on section 2.4.2.

3.2.1   Strong form of the problem

The strong form of the problem is given as follows: Given a scalar $k \in R$ and a vector field $\mathbf{b}$ defined on a region $\Omega \subset \mathbb{R}^{n_{sd}}$, find $\phi(x,t)$ such that:

$$
\begin{aligned}
\dot{\phi} - k\Delta\phi + \mathbf{b} \cdot \nabla\phi &= f & \text{on } \Omega \times I \\
\phi &= g & \text{on } \Gamma_D \times I \\
k\mathbf{n} \cdot \nabla\phi &= h & \text{on } \Gamma_N \times I \\
\phi(x,0) &= \phi^0 & \text{on } \Omega
\end{aligned}
\tag{3.4}
$$

where $\mathbf{n}$ is the vector normal to the boundary and Γ_D and Γ_N are the Dirichlet and Newman boundaries respectively, and we can state that $\partial\Omega = \Gamma = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \varnothing$.

3.2.2   Weak Formulation and Spatial Discretization

Weak Formulation.   Lets consider the following function spaces,

$$
S = \{\phi \in H^1(\Omega) : \phi = g \text{ on } \Gamma_D\}
\tag{3.5}
$$

$$
V = \{v \in H^1(\Omega) : v = 0 \text{ on } \Gamma_D\} = H_0^1
\tag{3.6}
$$

For this problem, we obtain the semi-discrete analog by multiplying (3.4) by $v \in V = H_0^1(\Omega)$

integrating over Ω and using the usual Green's formula. Then, we obtain the following varia-
tional formulation: Find $\phi(t) \in S$, $t \in I$ such that:

$$(\dot{\phi}(t), v) + a(\phi(t), v) \quad = \quad (f(t), v) + \int_{\Gamma_N} hv d\Gamma \quad \forall v \in V \quad t \in I \qquad (3.7)$$

with the bilinear operators defined as

$$
\begin{aligned}
a(\phi, v) &= \int_\Omega k \nabla \phi \cdot \nabla v d\Omega \\
b(\mathbf{b}; \phi, v) &= \int_\Omega (\mathbf{b} \cdot \nabla \phi) v d\Omega \\
(\phi, v) &= \int_\Omega \phi v d\Omega
\end{aligned}
$$

Galerkin formulation.   Now, let S_h be a finite dimensional subspace of S and V_h be a finite
dimensional subspace of V with basis $\{N_1, \dots, N_M\}$. Setting ϕ_h element of S_h and $v_h = v$
element of V_h, then the Galerkin variational formulation is defined as: Find $\phi_h(t) \in S_h$, $t \in I$
such that

$$(\dot{\phi}_h(t), v) + a(\phi_h(t), v) + b(\mathbf{b}; \phi, v) \quad = \quad (f(t), v) + \int_{\Gamma_N} hv d\Gamma \quad \forall v \in V_h \quad t \in I \qquad (3.8)$$

Finite element formulation.   We can choose to build the spaces S_h and V_h through finite
element approximations, such that

$$\bar{\Omega} = \bigcup_{e=1}^{n_{el}} \Omega_e \qquad \text{and,}$$

$$\bigcap_{e=1}^{n_{el}} \Omega_e = \varnothing$$

Here n_{el} is the number of elements and Ω_e is the element domain corresponding to the element
e.

3.2.3   Stabilized Finite Element Formulation

Following (TEZDUYAR; OSAWA, 2000), the stabilized finite element formulation of (3.4)
can be written as follows: Find $\phi_h \in S_h$ such that $\forall v_h \in V_h$:

$$
\int_\Omega v_h \left(\dot{\phi}_h(t) + \mathbf{b} \cdot \nabla \phi_h - f \right) d\Omega + \int_\Omega k \nabla \phi_h \cdot \nabla v_h
$$
$$
+ \sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau_{SUPG} \mathbf{b} \cdot \nabla v_h \left(\dot{\phi}_h(t) + \mathbf{b} \cdot \nabla \phi_h - k \nabla \cdot \nabla \phi_h - f \right) d\Omega = \int_{\Gamma_N} v_h h d\Gamma
$$
$$(3.9)$$

Here, τ_{SUPG} is the SUPG stabilization parameter. Alternately, we can express the previous
equation as follows. Let $w = v + \tau \, \mathbf{u} \cdot \nabla v$ where $v \in V_h$. Then equation (3.9) can be written as

follows: Find $\phi_h(t) \in S_h$, $t \in I$ such that $\forall v \in V_h$ and, $t \in I$,

$$(\dot{\phi}_h(t), w) + a(\phi_h(t), w) \;\; = \;\; (f(t), w) + (h, v)_{\Gamma_N} \tag{3.10}$$

3.2.4 Time discretization

Applying the generalized Crank-Nicholson time-discretization method, we obtain the following problem: Find $\phi_h^n \in V_h$, $n = 0, \ldots, N$ such that $\forall v \in V_h$,

$$\left(\frac{\phi_h^n - \phi_h^{n-1}}{\Delta t}, w \right) + a \left(\theta \phi_h^n + (1 - \theta) \phi_h^{n-1}, w \right) =$$
$$(\theta f(t_n) + (1 - \theta) f(t_{n-1}), w) + (h, v)_{\Gamma_N} \tag{3.11}$$

Reordering and taking advantage on the bilinearity of the operators we get,

$$(\phi_h^n, w) + \Delta t \theta a(\phi_h^n, w) = (\phi_h^{n-1}, w) - \Delta t (1 - \theta) a(\phi_h^{n-1}, w)$$
$$+ \Delta t \theta (f(t_n), w) + \Delta t (1 - \theta)(f(t_{n-1}), w) + \Delta t (h, v)_{\Gamma_N} \tag{3.12}$$

3.3 Navier-Stokes equations

3.3.1 Strong form of the problem

The strong form of the problem is given by extending equations (2.51) and (2.52) to include boundary and initial conditions. The governing equations and associated initial/boundary conditions are:

$$\rho_r(\phi) \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) + \nabla p - \frac{1}{Re} \nabla \cdot (2 \mu_r(\phi) \varepsilon(\mathbf{u})) = \mathbf{f} \qquad \text{on } \Omega \times I \qquad \text{(3.13a)}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{on } \Omega \times I \qquad \text{(3.13b)}$$

$$\mathbf{u} = \mathbf{u}_D \qquad \text{on } \Gamma_D \times I \qquad \text{(3.13c)}$$

$$\mathbf{n} \cdot \sigma = \mathbf{h} \qquad \text{on } \Gamma_N \times I \qquad \text{(3.13d)}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \qquad \text{on } \Omega \qquad \text{(3.13e)}$$

where we assume that $\nabla \cdot \mathbf{u}_0 = 0$, and the force term $\mathbf{f}$ is given by,

$$\mathbf{f} = \frac{\rho_r(\phi)}{Fr^2} \mathbf{g} + \frac{1}{We} k \nabla H_\Gamma \qquad \text{on } \Omega \times I \tag{3.14}$$

3.3.2   Weighting functions and trial solutions

The trial solution space $\mathbf{S}$ containing the approximating functions for the velocity is characterized as follows:

$$\mathbf{S} = \{\mathbf{v} \in (H^1(\Omega))^{n_{sd}} | \mathbf{v} = \mathbf{v}_D \text{ on } \Gamma_D\} \tag{3.15}$$

Note the candidate approximating functions must satisfy Dirichlet boundary conditions on Γ_D. The weighting funtions of the velocity w, belong to the space of functions $\mathbf{V}$. Elements of this class share the same characteristics as those in class $\mathbf{S}$, except that weighting functions are required to vanish on Γ_D where the velocity is prescribed. The weighting functions space $\mathbf{V}$ is defined by

$$\mathbf{V} = \{\mathbf{w} \in (H^1(\Omega))^{n_{sd}} | \mathbf{w} = 0 \text{ on } \Gamma_D\} \tag{3.16}$$

Additionally, we introduce the space of functions Q for pressure. Since no spatial derivatives of pressure appear on the weak form of the problem, functions in Q are simply required to be square-integrable. Consequently, we can define the pressure space of functions Q as,

$$Q = L_2(\Omega) \tag{3.17}$$

which suffices the trial solution space and the weighting function space for pressure.

3.3.3   Weak Formulation and Spatial Discretization

Weak Formulation.   Projecting equation (2.51) onto the space of weighting functions $\mathbf{w} \in \mathbf{V}$ for the momentum equation and $q \in Q$ for the incompressibility equation (2.52), we obtain the following variational formulation: given $\mathbf{f}$, $\mathbf{u}_D$, $\mathbf{h}$ and $\mathbf{u}_0$, find $\mathbf{u}(\mathbf{x}, t) \in \mathbf{S} \times I$ and $p(\mathbf{x}, t) \in Q \times I$, such that for all $(\mathbf{w}, q) \in \mathbf{V} \times Q$,

$$(\mathbf{w}, \mathbf{u}_t) + c(\mathbf{u}; \mathbf{w}, \mathbf{u}) + d(\mathbf{w}; p, \mathbf{u}) + e(q, \mathbf{u}) = (\mathbf{w}, \mathbf{f}) + (\mathbf{w}, \mathbf{h})_{\Gamma_N} \tag{3.18}$$

where the previous operators are defined as,

$$c(\mathbf{u}; \mathbf{w}, \mathbf{v}) = \int_\Omega \mathbf{w} \cdot \rho_r(\phi)(\mathbf{u} \cdot \nabla \mathbf{v}) d\Omega \tag{3.19}$$

$$d(\mathbf{w}; p, \mathbf{u}) = \int_\Omega \varepsilon(\mathbf{w}) : \sigma^*(p, \mathbf{u}) d\Omega \tag{3.20}$$

$$e(q, \mathbf{u}) = \int_\Omega q \nabla \cdot \mathbf{u} d\Omega \tag{3.21}$$

Galerkin Spatial Discretization.   To get a discrete analog for the previous formulation we need to introduce local approximations for both the velocity $\mathbf{u}^h$ and pressure p^h, as well as for their associated weighting functions $\mathbf{w}^h$ and q^h. We denote $\mathbf{S}^h$ and $\mathbf{V}^h$ the finite dimensional subspaces of $\mathbf{S}$ and $\mathbf{V}$, and Q^h the finite dimensional space of Q.

For the Galerkin formulation, the velocity approximation $\mathbf{u}^h \in \mathbf{S}^h$ admits the representation $\mathbf{u}^h = \mathbf{v}^h + \mathbf{u}_D^h$, where the field $\mathbf{u}_D^h$ satisfies the Dirichlet boundary condition. Thus, the auxiliary velocity $\mathbf{v}^h$ belongs to the same space as the test function $\mathbf{w}^h$, namely $\mathbf{V}^h$. Furthermore, the Galerkin spatial discretization proceeds as follows: For each $t \in I$, we define $\mathbf{u}_D^h(t) \in \mathbf{S}^h$ such that $\mathbf{u}^h(t) = \mathbf{v}^h(t) + \mathbf{u}_D^h(t)$. Then we seek the auxiliary velocity field $\mathbf{v}^h(\cdot, t) \in \mathbf{V}^h$ and pressure $p^h(\cdot, t) \in Q^h$, such that, for all $(\mathbf{w}^h, q^h) \in \mathbf{V}^h \times Q^h$,

$$
(\mathbf{w}^h, \mathbf{v}_t^h) + c(\mathbf{v}^h; \mathbf{w}^h, \mathbf{v}^h) + d(\mathbf{w}^h; p^h, \mathbf{v}^h) + e(q^h, \mathbf{v}^h) = (\mathbf{w}^h, \mathbf{f}^h) +
$$
$$
(\mathbf{w}^h, \mathbf{h}^h)_{\Gamma_N} - c(\mathbf{v}^h; \mathbf{w}^h, \mathbf{u}_D^h) - d(\mathbf{w}^h; p^h, \mathbf{u}_D^h) - e(q^h, \mathbf{u}_D^h) \tag{3.22}
$$

The next step in the Galerkin formulation consist in approximating the velocity components u_i^h in terms of shape functions and associated nodal values. The velocity components are approximated as follows:

$$
v_i^h(\mathbf{x}) = \sum_{A \in \eta \, \eta_{Di}} N_A(\mathbf{x}) v_{iA} \tag{3.23a}
$$

$$
u_{Di}^h(\mathbf{x}) = \sum_{A \in \eta_{Di}} N_A(x) u_{Di}(\mathbf{x}_A) \tag{3.23b}
$$

where η denotes the set of global velocity node numbers in the finite element mesh, $\eta_{Di} \subset \eta$ the subset of velocity nodes belonging to the Dirichlet portion of the boundary, N_A is the shape function associated with the global node number A, and v_{iA} the value of u_i^h at node number A. In the Galerkin formulation, test functions are defined such that,

$$
w_i^h \in V_i^h = \operatorname*{span}_{A \in \eta \, \eta_{Di}} \{N_A\} \tag{3.24}
$$

The pressure field is interpolated using a different set of pressure nodes denoted by $\bar{\eta}$ and the shape functions $\bar{N}_{\bar{A}}$, as,

$$
p^h(\mathbf{x}) = \sum_{\bar{A} \in \bar{\eta}} \bar{N}_{\bar{A}}(\mathbf{x}) p_{\bar{A}} \tag{3.25}
$$

where $\bar{A}$ is the global pressure node number and $p_{\bar{A}}$ is the pressure value at $\bar{A}$. Similarly, the

weighting function q^h for the pressure is expressed as,

$$q^h \in Q^h = \operatorname*{span}_{\bar{A} \in \bar{\eta}} \{\bar{N}_{\bar{A}}\} \tag{3.26}$$

Babuska-Brezzi compatibility condition.   In order to ensure the solvability of the discrete problem, the election of subspaces must be made while satisfying the following condition: There is a positive constant c such that,

$$\sup_{\mathbf{v} \in \mathbf{V}_h} \frac{(q, div\mathbf{v})}{\|\mathbf{v}\|_1} \geqslant c \|q\|_0 \quad \forall q \in Q_h \quad \mathbf{v} = (u, v) \tag{3.27}$$

$$\text{where} \qquad \|\mathbf{v}\|_1 = \sum_{i=1}^{n_{sd}} \left(\int_{\Omega} [u_i^2 + |\nabla u_i|^2] d\Omega \right)^{\frac{1}{2}}$$

$$\|q\|_0 = \left(\int_{\Omega} q^2 d\Omega \right)^{\frac{1}{2}}$$

Here, the inequality (3.27) is called the Babuska-Brezzi condition. Keeping this condition in mind, we are now able to select proper subspaces. A good choice of $\mathbf{V}^h$ and Q^h on square Ω_e elements is given by,

$$V^h = \{\mathbf{v} \in \mathbf{V} : \mathbf{v}|_{\Omega_e} \in Q_2(\Omega_e)\} \tag{3.28a}$$

$$H^h = \{q \in Q : q|_{\Omega_e} \in P_1(\Omega_e)\} \tag{3.28b}$$

where $Q_2(\Omega_e)$ is the set of biquadratic functions on Ω_e, and $P_1(\Omega_e)$ is the space of linear functions defined on Ω_e. This is,

$$Q_2(\Omega_e) = \{v : v(x) = \sum_{i,j=0}^{2} a_{ij} x_1^i x_2^j, x \in \Omega_e \text{ where } a_{ij} \in R\} \tag{3.29a}$$

$$P_r(\Omega_e) = \{p : p(x) = \sum_{i=0}^{r} a_i x^i, x \in \Omega_e \text{ where } a_i \in R\} \tag{3.29b}$$

Finally, the finite element discretization of this weak form yields the system of semi-discrete equations $t \in I$,

$$\mathbf{M}\dot{\mathbf{v}}(t) + [\mathbf{K} + \mathbf{C}(\mathbf{u}(t))]\mathbf{v}(t) + \mathbf{G}\mathbf{p}(t) = \mathbf{f}(t, \mathbf{u}(t)) \tag{3.30a}$$

$$\mathbf{G}^T \mathbf{v}(t) = \mathbf{h}(t) \tag{3.30b}$$

$$\mathbf{v}(0) = \mathbf{u}_0 - \mathbf{u}_D(0) \tag{3.30c}$$

where $\mathbf{M}$ is the standard finite element matrix, $\mathbf{K}$ is the viscosity matrix, $\mathbf{C}(\mathbf{u}(t))$ is the convection matrix and $\mathbf{G}$ and $\mathbf{G}^T$ are, respectively, the discrete gradient operator and the discrete divergence operator.

3.3.4 Stabilized Finite Element Formulation

Let us assume that we have some suitably defined finite dimensional trial solution and test function spaces for velocity and pressure: $\mathbf{S}^h$, $\mathbf{V}^h$, and Q^h. The stabilized finite element formulation of equation (3.13) can be written as follows: find $\mathbf{u}^h \in \mathbf{S}^h \times I$ and $p^h \in Q^h \times I$, for all $(\mathbf{w}^h, q^h) \in \mathbf{V}^h \times Q^h$, such that,

$$
\begin{aligned}
&\left(\mathbf{w}^h, \mathbf{u}_t^h\right) + c\left(\mathbf{u}^h; \mathbf{w}^h, \mathbf{u}^h\right) + d\left(\mathbf{w}^h; p^h, \mathbf{u}^h\right) + e\left(q^h, \mathbf{u}^h\right) - \left(\mathbf{w}^h, \mathbf{f}^h\right) - \left(\mathbf{w}^h, \mathbf{h}^h\right)_{\Gamma_N} \\
&+ \sum_{e=1}^{n_{el}} \left(\left[\tau_{\text{SUPG}}\mathbf{u}^h \cdot \nabla\mathbf{w}^h + \frac{1}{\rho_r(\phi)}\tau_{\text{PSPG}}\nabla q^h\right], \Re(\mathbf{u}_t^h, \mathbf{a}^h, p^h, \mathbf{u}^h)\right)_{\Omega_e} \\
&+ \sum_{e=1}^{n_{el}} \left(\tau_{\text{LSIC}}\nabla \cdot \mathbf{w}^h, \rho_r(\phi)\nabla \cdot \mathbf{u}^h\right)_{\Omega_e} = 0
\end{aligned}
\tag{3.31}
$$

where

$$
\Re(\mathbf{u}_t^h, \mathbf{a}^h, p^h, \mathbf{u}^h) = \rho_r(\phi)\left(\mathbf{u}_t^h + \mathbf{a}^h \cdot \nabla\mathbf{u}^h - \mathbf{f}^h\right) - \nabla \cdot \sigma^*(p^h, \mathbf{u}^h)
$$

is the residual of the momentum equation. Here the stabilization parameters are defined as follows, (TEZDUYAR; OSAWA, 2000)

$$
h_{UGN} = 2\|\mathbf{u}^h\| \left(\sum_{a=1}^{n_{en}} |\mathbf{u}^h \cdot \nabla N_a|\right)^{-1}
\tag{3.32a}
$$

$$
\tau_{SUGN1} = \frac{h_{UGN}}{2\|\mathbf{u}\|}
\tag{3.32b}
$$

$$
\tau_{SUGN2} = \frac{\Delta t}{2}
\tag{3.32c}
$$

$$
\tau_{SUGN3} = \frac{h_{UGN}^2}{4\nu}
\tag{3.32d}
$$

$$
\tau_{SUPG} = \left(\frac{1}{\tau_{SUGN1}^2} + \frac{1}{\tau_{SUGN2}^2} + \frac{1}{\tau_{SUGN3}^2}\right)^{-1/2}
\tag{3.32e}
$$

$$
\tau_{PSPG} = \tau_{SUPG}
\tag{3.32f}
$$

$$
\tau_{LSIC} = \frac{h_{UGN}}{2}\|\mathbf{u}^h\|z
\tag{3.32g}
$$

$$
z = \frac{Re_{UGN}}{3} \text{ for } Re_{UGN} \leqslant 3,\ z = 1 \text{ for } Re_{UGN} > 3
\tag{3.32h}
$$

$$
\text{with } Re_{UGN} = \frac{\|\mathbf{u}^h\|h_{UGN}}{2\nu}
\tag{3.32i}
$$

where N_a is the interpolation function associated with the node a.

3.3.5 Time discretization

Finite differences are employed for the time discretization. This is given by the θ-scheme. Equation (3.31) then become,

$$
\begin{aligned}
\left(\mathbf{w}^h, \frac{(\mathbf{u}_{n+1}^h - \mathbf{u}_n^h)}{\Delta t} \right) &+ c\left(\mathbf{u}_\alpha^h; \mathbf{w}^h, \mathbf{u}_\theta^h \right) + d\left(\mathbf{w}^h; p_{n+1}^h, \mathbf{u}_\theta^h \right) \\
&+ e\left(q^h, \mathbf{u}_\theta^h \right) - \left(\mathbf{w}^h, \mathbf{f}^h \right) - \left(\mathbf{w}^h, \mathbf{h}^h \right)_{\Gamma_N} \\
&+ \sum_{e=1}^{n_{el}} \left(\tau_{\text{SUPG}} \mathbf{u}_\alpha^h \cdot \nabla \mathbf{w}^h, \Re\left(\frac{(\mathbf{u}_{n+1}^h - \mathbf{u}_n^h)}{\Delta t}, \mathbf{u}_\alpha^h, p_{n+1}^h, \mathbf{u}_\theta^h \right) \right)_{\Omega_e} \\
&+ \sum_{e=1}^{n_{el}} \left(\frac{\tau_{\text{PSPG}} \nabla q^h}{\rho_r(\phi)}, \Re\left(\frac{(\mathbf{u}_{n+1}^h - \mathbf{u}_n^h)}{\Delta t}, \mathbf{u}_\alpha^h, p_{n+1}^h, \mathbf{u}_\theta^h \right) \right)_{\Omega_e} \\
&+ \sum_{e=1}^{n_{el}} \left(\tau_{\text{LSIC}} \nabla \cdot \mathbf{w}^h, \rho_r(\phi) \nabla \cdot (\mathbf{u}_\theta^h) \right)_{\Omega_e} = 0
\end{aligned}
\tag{3.33}
$$

The parameters θ and α are taken to be in the interval $[0,1]$ and we define $\mathbf{u}_\theta^h$ and $\mathbf{u}_\alpha^h$ as,

$$
\begin{aligned}
\mathbf{u}_\theta^h &= \theta \mathbf{u}_{n+1}^h + (1-\theta)\mathbf{u}_n^h \text{ and,} \\
\mathbf{u}_\alpha^h &= \alpha \mathbf{u}_{n+1}^h + (1-\alpha)\mathbf{u}_n^h
\end{aligned}
$$

Here, $\alpha = \theta = 0$ gives the explicit Euler method, $\theta = 1$, $\alpha = 0$ gives a semi-implicit method and $\alpha = \theta = 1$ an fully implicit Euler method. Also note that Crank-Nicholson, $\theta = 1/2$ is the only second-order accurate method.

In the resulting expansion of equation (3.33), we assume that $\mathbf{u}_{n+1}^h$ appearing in the advective operator $\tau_{\text{SUPG}} \mathbf{u}_{n+1}^h \cdot \nabla \mathbf{w}^h$ is evaluated at time level $n+1$ but nonlinear iteration level i rather than $i+1$. Note that this term only appears if the parameter α is different from 0.

3.3.6 Linearization

After time discretization, equation (3.33) can be written in residual form for the unknown nodal values $\mathbf{U}_{n+1}$ as the nonlinear algebraic system,

$$
R(\mathbf{U}_{n+1}) = 0
\tag{3.34}
$$

We can see now, that for any $\alpha \neq 0$ we need to solve a nonlinear system. To do so, we use a nonlinear Newton solver. As we are solving the pressure-velocity coupled system,

we can select LBB stable elements, then being able to set the τ_{PSPG} parameter equal to 0. For our problem, we choose quadrilateral elements with 8 degrees of freedom, with second order accuracy for the velocity and first order accuracy for the pressure.

The goal is to define a sequence of linear problems that, when solved, converge to obtain the solution $\mathbf{U}_{n+1}$ of the nonlinear system (3.34). We can achieve this goal with the Newton-Raphson method, that results from expanding (3.34) with a Taylor series about iterate $\mathbf{U}_{n+1}^l$:

$$\left[\frac{\partial \mathbf{R}(\mathbf{U}_{n+1}^l)}{\partial \mathbf{U}_{n+1}}\right] \delta \mathbf{U}_{n+1}^{l+1} = -\mathbf{R}(\mathbf{U}_{n+1}^l) + \mathcal{O}\left((\delta \mathbf{U}_{n+1}^{l+1})^2\right) \tag{3.35}$$

where $\frac{\partial \mathbf{R}}{\partial \mathbf{U}}$ is the Jacobian matrix for the nonlinear system $\delta \mathbf{U}_{n+1}^{l+1} = \mathbf{U}_{n+1}^{l+1} - \mathbf{U}_{n+1}^l$. Setting $\mathcal{O}\left((\delta \mathbf{U}_{n+1}^{l+1})^2\right) \approx 0$ yields Newton's method

$$\left[\frac{\partial \mathbf{R}(\mathbf{U}_{n+1}^l)}{\partial \mathbf{U}_{n+1}}\right] \delta \mathbf{U}_{n+1}^{l+1} = -\mathbf{R}(\mathbf{U}_{n+1}^l) \tag{3.36}$$

that results in an implicit linear system for $\delta \mathbf{U}_{n+1}^{l+1}$ and a sequence of iterates $l = (0,\ 1,\ \dots)$ which converges to $\mathbf{U}_{n+1}$ (KIRK, 2007). Note that the left hand side of equation (3.36) is nothing more than the derivative of $\mathbf{R}$ in the direction specified by $\delta \mathbf{U}$, which is defined by

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right] \delta \mathbf{U} = \lim_{\epsilon \to 0}\left\{\frac{\mathbf{R}(\mathbf{U} + \epsilon \delta \mathbf{U}) - \mathbf{R}(\mathbf{U})}{\epsilon}\right\} \tag{3.37}$$

It's worthwhile to recall that Newton's method exhibits second-order conditional convergence, meaning that the magnitude of the residual decreases quadratically at successive iterates provided that the initial guess is sufficiently close to the unknown. For this reason, our algorithm adopts linear extrapolation, using previous solutions to estimate the initial Newton's iterate. This has shown marked improvement on the convergence rate, but even with this implementation, sometimes the initial guess is far from the exact solution and the algorithm diverges. Our algorithm deals with this problem by reducing the time step, therefore looking for the unknown closer to the previous solution.

Solving a large system of equations such as the one presented by equation (3.34) can be a hard task to the linear solver, as it can lead to a bad-conditioned system of equations. An approximate block ILU factorization is employed.

CHAPTER **4**

NUMERICAL IMPLEMENTATION

In this chapter we describe the numerical implementation of the formulations detailed on previous chapters. The numerical finite element simulation code was written in C++ language with the aid of the *libmesh* library (KIRK et al., 2006) as base tool.

libmesh is a parallel adaptive finite element library for simulating partial differential equations. Among many advantages this library offers the possibility of selecting different generic 1D, 2D and 3D finite element types and interpolation functions; runtime selection of different quadrature rules; mesh creation, modification, input, output and format translation utilities; with support for adaptive mesh refinement (AMR) computations in parallel platforms. With this tool, the program implementation became much easier and was performed in short period of time.

4.1 Solution Methodology

Assuming that the initial level set function ϕ_0, the initial pressure and the initial velocity field are known, the algorithm can be summarized as follows,

1. Refine the mesh in the region close to the interface.

2. Calculate the Heaviside function from equation (2.59).

3. Calculate Curvature as in section 2.4.3.

4. Solve discrete Navier-Stokes equations (2.51) and (2.52) through a Newton nonlinear iterative procedure to obtain the new velocity field **u**. Solve the symmetric linear systems with an ILU preconditioned GMRES method. Refine and coarsen the mesh if needed.

5. Solve equation (2.53) to advect the level set function ϕ by the velocity field $\mathbf{u}$.

6. Reinitialize the level set function by solving equation (2.62) with an iterative procedure to restore the distance function property in the region close to the zero level set.

7. Save results, return to step 2)

4.2 Convection-Diffusion Problem

4.2.1 System Assembly

The assembly of the element contributions given by equation (3.12) into the complete system, results in a matrix equation of the form

$$\mathbf{KU} = \mathbf{F} \tag{4.1}$$

where $\mathbf{U}$ is the vector of the unknown nodal values. In practice, the global matrix $\mathbf{K}$, and nodal vector $\mathbf{F}$, result from the topological assembly of element contributions. The addition of these contributions to the appropriate location in the global matrix, $\mathbf{K}$, and nodal vector $\mathbf{F}$, can be represented through the action of an assembly operator $\mathbb{A}^e$ acting on the local element matrix and nodal vector as follows:

$$\mathbf{K} = \mathbb{A}^{\mathbf{e}}\mathbf{K}^{\mathbf{e}}, \qquad K_{ab}^e = \int_{\Omega e} N_a N_b d\Omega$$
$$+ \int_{\Omega_e} \theta \Delta t (\nabla N_b \cdot \mathbf{b}) N_a d\Omega$$
$$+ \int_{\Omega_e} \epsilon \theta \Delta t (\nabla N_a \cdot \nabla N_b) d\Omega$$
$$+ \int_{\Omega_e} \tau_{\text{SUPG}} (\nabla N_a \cdot \mathbf{b})(N_b - \epsilon \theta \Delta t (\nabla \cdot \nabla N_b) + \mathbf{b} \cdot \nabla N_b) d\Omega$$

$$\mathbf{F} = \mathbb{A}^{\mathbf{e}}\mathbf{F}^{\mathbf{e}}, \qquad F_a^e = \int_{\Omega e} \phi_{n-1}^h N_a d\Omega$$
$$- \int_{\Omega_e} \Delta t (1 - \theta)(\nabla \phi_{n-1}^h \cdot \mathbf{b}) N_a d\Omega$$
$$- \int_{\Omega_e} \epsilon (1 - \theta) \Delta t (\nabla N_a \cdot \nabla \phi_{n-1}^h) d\Omega$$
$$+ \int_{\Omega_e} \Delta t f N_a d\Omega$$
$$+ \int_{\Omega_e} \tau_{\text{SUPG}} (\nabla N_a \cdot \mathbf{b})(\phi_{n-1}^h + \Delta t(f) + \epsilon \Delta t (1 - \theta) \nabla \cdot \nabla \phi_{n-1}^h$$
$$- \Delta t (1 - \theta)(\mathbf{b} \cdot \nabla \phi_{n-1}^h)) d\Omega$$

where N_a is the shape function associated with the node number a of the element Ω^e in the finite element mesh. Note that the approximation ϕ^h can also be written,

$$\phi^h(\mathbf{x}) = \sum_{a \in \eta_{\Omega^e}} N_a(\mathbf{x}) U_a \qquad (4.2)$$

where η_{Ω^e} is the set of nodes belonging to the element Ω^e.

4.3 Navier-Stokes equations

4.3.1 System Assembly

Following the procedure described in the previous section, the assembly of the element contributions for the Navier Stokes equations given by equation (3.36), into the complete system results in a matrix equation of the form

$$\mathbf{KU} = \mathbf{F} \qquad (4.3)$$

where $\mathbf{U}$ is the vector of the unknown nodal values. In practice, the global matrix $\mathbf{K}$, and nodal vector $\mathbf{F}$, result from the topological assembly of element contributions. The addition of these contributions to the appropriate location in the global matrix, $\mathbf{K}$, and nodal vector $\mathbf{F}$, can be represented through the action of an assembly operator $\mathbb{A}^e$ acting on the local element matrix and nodal vector as follows:

$$\mathbf{K} = \mathbb{A}^e \mathbf{K^e} \qquad (4.4)$$

$$\mathbf{F} = \mathbb{A}^e \mathbf{F^e} \qquad (4.5)$$

where again, we can split the element matrix $\mathbf{K^e}$ and element vector $\mathbf{F^e}$ into submatrices each corresponding to the unknown nodal values of the velocity components u and v (assuming $n_{sd} = 2$) and pressure p, as follows,

$$K^e = \begin{bmatrix} K_{uu}^e & K_{uv}^e & K_{up}^e \\ K_{vu}^e & K_{vv}^e & K_{vp}^e \\ K_{pu}^e & K_{pv}^e & K_{pp}^e \end{bmatrix} \qquad F^e = \begin{bmatrix} F_u^e \\ F_v^e \\ F_p^e \end{bmatrix} \qquad (4.6)$$

Note that we can also write equation (3.33) as a sum of terms as follows,

$$\left(\mathbf{w}^h, \frac{(\mathbf{u}_{n+1}^h - \mathbf{u}_n^h)}{\Delta t} \right) \qquad \text{TEMPORAL}$$

$$+ c\left(\mathbf{u}_\alpha^h; \mathbf{w}^h, \mathbf{u}_\theta^h \right) \qquad \text{CONVECTION}$$

$$+ d\left(\mathbf{w}^h; p_{n+1}^h, \mathbf{u}_\theta^h\right) \qquad\qquad \text{STRESS}$$

$$+ e\left(q^h, \mathbf{u}_\theta^h\right) + \sum_{e=1}^{n_{el}} \left(\tau_{\text{LSIC}} \nabla \cdot \mathbf{w}^h, \rho_r(\phi) \nabla \cdot (\mathbf{u}_\theta^h)\right)_{\Omega_e} \qquad \text{CONTINUITY}$$

$$- \left(\mathbf{w}^h, \mathbf{f}^h\right) - \left(\mathbf{w}^h, \mathbf{h}^h\right)_{\Gamma_N} \qquad\qquad \text{FORCE}$$

$$+ \sum_{e=1}^{n_{el}} \left(\tau_{\text{SUPG}} \mathbf{u}_\alpha^h \cdot \nabla \mathbf{w}^h + \frac{\tau_{\text{PSPG}} \nabla q^h}{\rho_r(\phi)}, \rho_r(\phi) \frac{(\mathbf{u}_{n+1}^h - \mathbf{u}_n^h)}{\Delta t}\right)_{\Omega_e} \quad \text{TEMPORAL}_{\text{SUPGPSPG}}$$

$$+ \sum_{e=1}^{n_{el}} \left(\tau_{\text{SUPG}} \mathbf{u}_\alpha^h \cdot \nabla \mathbf{w}^h + \frac{\tau_{\text{PSPG}} \nabla q^h}{\rho_r(\phi)}, \rho_r(\phi) \mathbf{u}_\alpha^h \cdot \nabla \mathbf{u}_\theta^h\right)_{\Omega_e} \quad \text{CONVECTION}_{\text{SUPGPSPG}}$$

$$+ \sum_{e=1}^{n_{el}} \left(\tau_{\text{SUPG}} \mathbf{u}_\alpha^h \cdot \nabla \mathbf{w}^h + \frac{\tau_{\text{PSPG}} \nabla q^h}{\rho_r(\phi)}, -\nabla \cdot \sigma^*(p_{n+1}^h, \mathbf{u}_\theta^h)\right)_{\Omega_e} \quad \text{STRESS}_{\text{SUPGPSPG}}$$

$$+ \sum_{e=1}^{n_{el}} \left(\tau_{\text{SUPG}} \mathbf{u}_\alpha^h \cdot \nabla \mathbf{w}^h + \frac{\tau_{\text{PSPG}} \nabla q^h}{\rho_r(\phi)}, -\rho_r(\phi) \mathbf{f}^h\right)_{\Omega_e} \quad \text{FORCE}_{\text{SUPGPSPG}}$$

$$= 0 \qquad\qquad\qquad (4.7)$$

Then, assuming $\tau_{\text{PSPG}} = \tau_{\text{LSIC}} = 0$, $\theta = 1$, $\alpha = 1$ (see Appendix A.1 for the general case), we can calculate each element submatrix writing the terms in the same order as follows,

$$
\begin{aligned}
Fu_a^e = \int_\Omega (& \\
& + (\rho_r u_n N_a) / (\Delta t) \\
& + 0 \\
& + \rho_r \left(\frac{\partial u_{n+1}^i}{\partial x} u + \frac{\partial u_{n+1}^i}{\partial y} v\right) N_a \\
& + 0 \\
& + \frac{k}{We} \frac{\partial H}{\partial x} N_a \\
& + \left(\rho_r \tau_{\text{SUPG}} u_n \left(\left(\frac{\partial N_a}{\partial x}\right) u + \left(\frac{\partial N_a}{\partial y}\right) v\right)\right) / (\Delta t) \\
& + 0 \\
& + \rho_r \tau_{\text{SUPG}} \left(\frac{\partial u_{n+1}^i}{\partial x} u + \frac{\partial u_{n+1}^i}{\partial y} v\right) \left(\left(\frac{\partial N_a}{\partial x}\right) u + \left(\frac{\partial N_a}{\partial y}\right) v\right) \\
& + \frac{k}{We} \frac{\partial H}{\partial x} \tau_{\text{SUPG}} \left(\left(\frac{\partial N_a}{\partial x}\right) u + \left(\frac{\partial N_a}{\partial y}\right) v\right) \\
&) \, d\Omega
\end{aligned}
$$

$$Fv_a^e = \int_\Omega ($$

$$+ \left(\rho_r v_n N_a \right) / \left(\Delta t \right)$$

$$+0$$

$$+\rho_r \left(\frac{\partial v_{n+1}^i}{\partial x} u + \frac{\partial v_{n+1}^i}{\partial y} v \right) N_a$$

$$+0$$

$$+ \left(\frac{k}{We} \frac{\partial H}{\partial y} + \frac{\rho_r}{Fr^2} \right) N_a$$

$$+ \left(\rho_r \tau_{\text{SUPG}} \left(\left(\frac{\partial N_a}{\partial x} \right) u + \left(\frac{\partial N_a}{\partial y} \right) v \right) v_n \right) / \left(\Delta t \right)$$

$$+0$$

$$+\rho_r \tau_{\text{SUPG}} \left(\frac{\partial v_{n+1}^i}{\partial x} u + \frac{\partial v_{n+1}^i}{\partial y} v \right) \left(\left(\frac{\partial N_a}{\partial x} \right) u + \left(\frac{\partial N_a}{\partial y} \right) v \right)$$

$$+ \left(\frac{k}{We} \frac{\partial H}{\partial y} + \frac{\rho_r}{Fr^2} \right) \tau_{\text{SUPG}} \left(\left(\frac{\partial N_a}{\partial x} \right) u + \left(\frac{\partial N_a}{\partial y} \right) v \right)$$

$$\left. \right) d\Omega$$

$$Kuu_{ab}^e = \int_\Omega ($$

$$+ \left(\rho_r N_b N_a \right) / \left(\Delta t \right)$$

$$+ \left(\left(2 \left(\frac{\partial N_b}{\partial x} \right) \left(\frac{\partial N_a}{\partial x} \right) + \left(\frac{\partial N_b}{\partial y} \right) \left(\frac{\partial N_a}{\partial y} \right) \right) \mu_r \right) / Re$$

$$+\rho_r \left(\left(\frac{\partial N_b}{\partial x} \right) u + \frac{\partial u_{n+1}^i}{\partial x} N_b + \left(\frac{\partial N_b}{\partial y} \right) v \right) N_a$$

$$+0$$

$$+0$$

$$+ \left(\rho_r \tau_{\text{SUPG}} N_b \left(\left(\frac{\partial N_a}{\partial x} \right) u + \left(\frac{\partial N_a}{\partial y} \right) v \right) \right) / \left(\Delta t \right)$$

$$+ - \left(\left(\left(\frac{\partial^2 N_b}{\partial x^2} + \frac{\partial^2 N_b}{\partial y^2} \right) \mu_r \tau_{\text{SUPG}} \left(\left(\frac{\partial N_a}{\partial x} \right) u + \left(\frac{\partial N_a}{\partial y} \right) v \right) \right) / Re \right)$$

$$+\rho_r \tau_{\text{SUPG}} \left(\left(\frac{\partial N_b}{\partial x} \right) u + \frac{\partial u_{n+1}^i}{\partial x} N_b + \left(\frac{\partial N_b}{\partial y} \right) v \right) \left(\left(\frac{\partial N_a}{\partial x} \right) u + \left(\frac{\partial N_a}{\partial y} \right) v \right)$$

$$+0$$

$$\left. \right) d\Omega$$

$$Kuv_{ab}^e = \int_\Omega ($$

$$+0$$

$$+ \left(\left(\frac{\partial N_b}{\partial x} \right) \left(\frac{\partial N_a}{\partial y} \right) \mu_r \right) / Re$$

$$+\frac{\partial u^i_{n+1}}{\partial y}\rho_r N_b N_a$$

$$+0$$

$$+0$$

$$+0$$

$$+0$$

$$+\frac{\partial u^i_{n+1}}{\partial y}\rho_r \tau_{\text{SUPG}}\left(\left(\frac{\partial N_a}{\partial x}\right)u+\left(\frac{\partial N_a}{\partial y}\right)v\right)N_b$$

$$+0$$

$$\left.\right) d\Omega$$

$$Kvu^e_{ab}=\int_\Omega ($$

$$+0$$

$$+\left(\left(\frac{\partial N_b}{\partial y}\right)\left(\frac{\partial N_a}{\partial x}\right)\mu_r\right)/Re$$

$$+\frac{\partial v^i_{n+1}}{\partial x}\rho_r N_b N_a$$

$$+0$$

$$+0$$

$$+0$$

$$+0$$

$$+\frac{\partial v^i_{n+1}}{\partial x}\rho_r \tau_{\text{SUPG}}N_b\left(\left(\frac{\partial N_a}{\partial x}\right)u+\left(\frac{\partial N_a}{\partial y}\right)v\right)$$

$$+0$$

$$\left.\right) d\Omega$$

$$Kvv^e_{ab}=\int_\Omega ($$

$$+\left(\rho_r N_b N_a\right)/\left(\Delta t\right)$$

$$+\left(\left(\left(\frac{\partial N_b}{\partial x}\right)\left(\frac{\partial N_a}{\partial x}\right)+2\left(\frac{\partial N_b}{\partial y}\right)\left(\frac{\partial N_a}{\partial y}\right)\right)\mu_r\right)/Re$$

$$+\rho_r\left(\left(\frac{\partial N_b}{\partial x}\right)u+\left(\frac{\partial N_b}{\partial y}\right)v+\frac{\partial v^i_{n+1}}{\partial y}N_b\right)N_a$$

$$+0$$

$$+0$$

$$+\left(\rho_r \tau_{\text{SUPG}}\left(\left(\frac{\partial N_a}{\partial x}\right)u+\left(\frac{\partial N_a}{\partial y}\right)v\right)N_b\right)/\left(\Delta t\right)$$

$$+ - \left(\left(\left(\frac{\partial^2 N_b}{\partial x^2} + \frac{\partial^2 N_b}{\partial y^2} \right) \mu_r \tau_{\text{SUPG}} \left(\left(\frac{\partial N_a}{\partial x} \right) u + \left(\frac{\partial N_a}{\partial y} \right) v \right) \right) / Re \right)$$

$$+ \rho_r \tau_{\text{SUPG}} \left(\left(\frac{\partial N_a}{\partial x} \right) u + \left(\frac{\partial N_a}{\partial y} \right) v \right) \left(\left(\frac{\partial N_b}{\partial x} \right) u + \left(\frac{\partial N_b}{\partial y} \right) v + \frac{\partial v_{n+1}^i}{\partial y} N_b \right)$$

$$+ 0$$

$$\left. \right) d\Omega$$

$$Kup_{ab}^e = \int_\Omega ($$

$$+ 0$$

$$+ - \left(\left(\frac{\partial N_a}{\partial x} \right) M_b \right)$$

$$+ 0$$

$$+ 0$$

$$+ 0$$

$$+ 0$$

$$+ \left(\frac{\partial M_b}{\partial x} \right) \tau_{\text{SUPG}} \left(\left(\frac{\partial N_a}{\partial x} \right) u + \left(\frac{\partial N_a}{\partial y} \right) v \right)$$

$$+ 0$$

$$+ 0$$

$$\left. \right) d\Omega$$

$$Kvp_{ab}^e = \int_\Omega ($$

$$+ 0$$

$$+ - \left(\left(\frac{\partial N_a}{\partial y} \right) M_b \right)$$

$$+ 0$$

$$+ 0$$

$$+ 0$$

$$+ 0$$

$$+ \left(\frac{\partial M_b}{\partial y} \right) \tau_{\text{SUPG}} \left(\left(\frac{\partial N_a}{\partial x} \right) u + \left(\frac{\partial N_a}{\partial y} \right) v \right)$$

$$+ 0$$

$$+ 0$$

$$\left. \right) d\Omega$$

$$Kpu_{ab}^e = \int_\Omega ($$

$$+0$$
$$+0$$
$$+0$$
$$+\left(\frac{\partial N_b}{\partial x}\right)M_a$$
$$+0$$
$$+0$$
$$+0$$
$$+0$$
$$+0$$
$$\Big)\,d\Omega$$

$$Kpv_{ab}^e = \int_\Omega (\,$$

$$+0$$
$$+0$$
$$+0$$
$$+\left(\frac{\partial N_b}{\partial y}\right)M_a$$
$$+0$$
$$+0$$
$$+0$$
$$+0$$
$$+0$$
$$\Big)\,d\Omega$$

$$Kpp_{ab}^e = 0$$

4.4 Boundary Conditions

In this work, the penalty method was used to impose boundary conditions. This method has the advantage of easily introducing constrains while maintaining the shape of the matrix. Suppose we wish to solve our standard matrix problem,

$$\mathbf{KU} = \mathbf{F} \tag{4.8}$$

where, as usual, $\mathbf{K}$ is symmetric and positive-definite, subject to a constraint on one of the

degrees of freedom, namely,

$$U_Q = q \tag{4.9}$$

where the subscript Q represents the equation number in the global ordering and q is a given constant. The modified problem consisting of (4.8) and (4.9) may be formulated as a *constrained variational problem*.

One way to impose boundary conditions is to use the **penalty method**. Suppose that node 8 is on Γ_D, so that $U_8 = q_8$ must be satisfied from the system of linear equations (4.8) we have,

$$K_{81}U_1 + K_{82}U_2 + \ldots + K_{88}U_8 + \ldots + K_{8N}U_N = F_8 \tag{4.10}$$

in addition to the boundary condition. If (4.10) is rewritten as

$$K_{81}U_1 + K_{82}U_2 + \ldots + (K_{88} + k)U_8 + \ldots + K_{8N}U_N = F_8 + kq_8 \tag{4.11}$$

by adding the term kU_8 on the left side and kq_8 on the right side, and if $k \to \infty$, it is clear that $U_8 \to q_8$ and the relation (4.10) is approximately equivalent to the boundary condition.

4.5   Adaptive Mesh Refinement

In this work, we make use of adaptive mesh refinement techniques to achieve higher accuracy with lower computational costs. The refinement of the mesh is performed based on an error estimation, that is initially set to be high on the region of the interface of both fluids, and subsequently calculated from the velocity vector and pressure coming from the solution of the Navier-Stokes calculation. This error estimation is calculated through the the techniques presented in (KELLY et al., 1983) in which the error is based on the jump of the gradients across inter-element boundaries.

4.6   Solution Algorithm

The algorithms described below combine the methodologies outlined in the previous sections.

4.6.1   Main Algorithm

Algorithm 4.1 Main Algorithm

 1: Initialize mesh and simulation parameters
 2: Set $\phi^n = \phi_0$
 3: Refine the mesh only on the region near the interface
 4: Set $\phi^n = \phi_0$ on new finer grid
 5: **for** $i = 1$ to $N_{timesteps}$ **do**
 6:      Calculate time step Δt (when using variable time steps)
 7:      Solve reinitialization function (see Algorithm 4.2)
 8:      Calculate Heaviside function $H_\epsilon(\phi)$ and Curvature k (see Algorithm 4.4)
 9:      Solve Navier Stokes nonlinear solver (see Algorithm 4.5)
10:      Solve Level Set advection equation (see Algorithm 4.3)
11:      Save Results
12: **end for**

4.6.2   Reinitialization Algorithm

See section 2.4.2 for further details.

Algorithm 4.2 Reinitialization Algorithm

 1: Given ϕ^n, $\Delta\tau$, tol_{reinit}
 2: Set $\varphi^n = \phi^n$
 3: Set $k_{reinit} \approx 0$, $k_{reinit} > 0$
 4: Set $\theta_{reinit} = 1$
 5: Define $\mathbf{w} = S(\varphi^{n-1}, h)\frac{\nabla\varphi^{n-1}}{|\nabla\varphi^{n-1}|}$
 6: Define $f = S(\varphi^{n-1}, h)$
 7: **repeat**
 8:      Set $\varphi^{n-1} = \varphi^n$
 9:      Set $\widetilde{\varphi}^n = \text{CONVECTIONDIFFUSION}(\mathbf{w}, f, \varphi^{n-1}, \Delta\tau, k_{reinit}, \theta_{reinit})$
10:      Calculate $\zeta = \lambda\delta_\epsilon(\widetilde{\varphi})|\nabla\widetilde{\varphi}|$
11:      Set $\varphi^n = \widetilde{\varphi}^n + \zeta$
12: **until** $\|\varphi^n - \varphi^{n-1}\|_\infty < tol_{reinit}h_{min}$
13: Set $\phi^{n-1} = \varphi^n$

4.6.3   Level Set Advection.

See section 2.4.1 for further details.

Algorithm 4.3 Level Set Advection Algorithm

1: Given ϕ^{n-1}, $\mathbf{u}$, Δt
2: Set $k_{lvlset} \approx 0$, $k > 0$
3: Set $\theta_{lvlset} = 0.5$
4: Define $f = 0$
5: Set $\phi^n = \text{CONVECTIONDIFFUSION}(\mathbf{u}, f, \phi^{n-1}, \Delta t, k_{lvlset}, \theta_{lvlset})$

4.6.4 Heaviside and Curvature Calculation

See sections 2.4 and 2.4.3 for further details.

Algorithm 4.4 Heaviside and Curvature Calculation

1: Given $\varphi = \varphi^n$, ϵ
2: Calculate Heaviside: $H_\epsilon(\varphi) = \left\{ \begin{array}{cc} 0 & \varphi < -\epsilon \\ \frac{1}{2} + \frac{\varphi}{2\epsilon} + \frac{1}{2\pi} sin\left(\frac{\pi\varphi}{\epsilon}\right) & -\epsilon \leq \varphi \leq \epsilon \\ 1 & \epsilon < \varphi \end{array} \right\}$
3: Set $nx_0 = 0$
4: Set $\Delta T \approx h$
5: Set $k_{nx} \approx 0$, $k > 0$
6: Set $\theta_{nx} = 1$
7: Define $\mathbf{v} = S(\varphi, h)\nabla\varphi$
8: Define $f_{nx} = \frac{1}{\Delta T}\frac{\partial\varphi}{\partial x}$
9: Set $nx = \text{CONVECTIONDIFFUSION}(\mathbf{v}, f_{nx}, 0, \Delta T, k_{nx}, \theta_{nx})$
10: Define $f_{ny} = \frac{1}{\Delta T}\frac{\partial\varphi}{\partial y}$
11: Set $ny = \text{CONVECTIONDIFFUSION}(\mathbf{v}, f_{ny}, 0, \Delta T, k_{nx}, \theta_{nx})$
12: Define $\mathbf{n} = \frac{(nx,ny)}{\|(nx,ny\|}$
13: Define $\mathbf{n}_\perp = (-ny, nx)$
14: Calculate Curvature: $k = -ny(\mathbf{n}_\perp \cdot \nabla nx) + nx(\mathbf{n}_\perp \cdot \nabla ny)$

4.6.5 Navier Stokes Algorithm

See sections 2.2.3, 2.2.4, 2.2.5, 2.3, 3.3 and specially 4.3 for further details.

Algorithm 4.5 Navier Stokes

 1: Set $\mathbf{U}_{n-2} = \mathbf{U}_{n-1}$ (Store older solution)
 2: Set $\mathbf{U}_{n-1} = \mathbf{U}_n$ (Store old solution)
 3: Set $\mathbf{w} = (0.5, 0.5, 1)$ (Refinement weights)
 4: Given tol
 5: Set $l = 0$
 6: Set *refineflag = true*
 7: **repeat**
 8:     $l = l + 1$
 9:     Set $\mathbf{U}^{l-1} = \mathbf{U}^l$ (Store previous nonlinear iteration)
10:     Assembly the system matrix $\mathbf{K} = \mathbb{A}^{\mathbf{e}}\mathbf{K}^{\mathbf{e}}$ described in section 4.3.1
11:     Assembly the system vector $\mathbf{F} = \mathbb{A}^{\mathbf{e}}\mathbf{F}^{\mathbf{e}}$ described in section 4.3.1
12:     Solve the linear system $\mathbf{K}\mathbf{U}^l = \mathbf{F}$
13:     Evaluate $\delta\mathbf{U} = \mathbf{U}^l - \mathbf{U}^{l-1}$
14:     **if** *refineflag* $==$ true **then**
15:         *refineflag* = REFINEMESH$(\mathbf{U}^l, \mathbf{w})$
16:     **end if**
17: **until** $\frac{\|\delta\mathbf{U}\|}{\|\mathbf{U}^l\|} < tol$ **or** $l == N_{\text{nlsteps}}$

4.6.6 Convection Diffusion Algorithm

See sections 3.2 and 4.2 for further details.

Algorithm 4.6 Convection Diffusion Function

 1: **function** CONVECTIONDIFFUSION$(\mathbf{b}, f, \phi_0, \Delta t, \epsilon, \theta)$
 2:     Using $\mathbf{b}, f, \phi_0, \Delta t, \epsilon, \theta$
 3:     Assembly the system matrix $\mathbf{K} = \mathbb{A}^{\mathbf{e}}\mathbf{K}^{\mathbf{e}}$ described in section 4.2.1
 4:     Assembly the system vector $\mathbf{F} = \mathbb{A}^{\mathbf{e}}\mathbf{F}^{\mathbf{e}}$ described in section 4.2.1
 5:     Solve the linear system $\mathbf{K}\mathbf{U} = \mathbf{F}$
 6:     **return** $\mathbf{U}$
 7: **end function**

4.6.7 Refinement Algorithm

See section 4.5 for further details.

Algorithm 4.7 Refinement Algorithm

 1: **function** REFINEMENT$(\mathbf{U}^l, \mathbf{w})$
 2:     Using $\mathbf{U}^l, \mathbf{w}$
 3:     Given $tol_{ref}, tol_{coarsen}$
 4:     Refine elements with error higher than tol_{ref}
 5:     Coarsen elements with error lower than $tol_{coarsen}$
 6: **end function**

CHAPTER 5

VALIDATION AND RESULTS

To validate the code and method and assess robustness and precision, we perform a number of test calculations. First, we start by running single fluid simulations to evaluate the Navier-Stokes solver accuracy. Next, we evaluate the pressure inside a static bubble and frequency of oscillation of an elliptical bubble and compare the results with analytical solutions. Then, we evaluate the rising velocity of a single bubble in a continuous phase for both low and moderate Reynolds numbers.

5.1 Navier Stokes Solver

5.1.1 Plane Poiseuille Flow

A validation of the balance of viscous and pressure terms in the Navier-Stokes solver is performed by evaluating the pressure gradient on a steady laminar viscous flow in a two-dimensional channel, known as plane Poiseuille flow. The viscous fluid flows from a region of higher pressure to one of lower pressure between two parallel plates separated by a fixed distance h.

The domain length is $8h$. Boundary conditions are non-slip at the top and bottom walls, prescribed velocity at the inflow on the left boundary, and prescribed pressure at the outflow on the right boundary. Under these conditions, considering steady developed flow, the pressure gradient is given by

$$\frac{\partial p}{\partial x} = -12\frac{\mu U}{h^2} \tag{5.1}$$

where U is the average velocity, and $\frac{\partial p}{\partial x}$ is the pressure gradient. Simulation parameters and results are listed in table 5.1.

Plane Poiseuille flow					
Physical Parameters					
Reference density	ρ_∞	kg/m^3	1000	1000	1000
Reference viscosity	μ_∞	Pa·s	0.001	0.001	0.0001
Average Velocity	U	m/s	0.001	0.01	0.01
Reference Length	h	m	0.001	0.001	0.001
Non-dimensional Numbers					
Reynolds Number	Re	$\frac{\rho U h}{\mu}$	1	10	100
Results					
Theoretical Pressure Gradient	$\frac{\partial p}{\partial x}$	Pa/m	-12	-120	-12
Pressure Gradient	$\frac{\partial p}{\partial x}$	Pa/m	-12.0010	-120.0210	-11.9872
Pressure Gradient relative error	$\frac{\|E_p\|}{\|\nabla p\|}$		$8.33 \cdot 10^{-5}$	$1.75 \cdot 10^{-4}$	$1.06 \cdot 10^{-3}$

Table 5.1: Pressure gradient evaluation parameters and results.
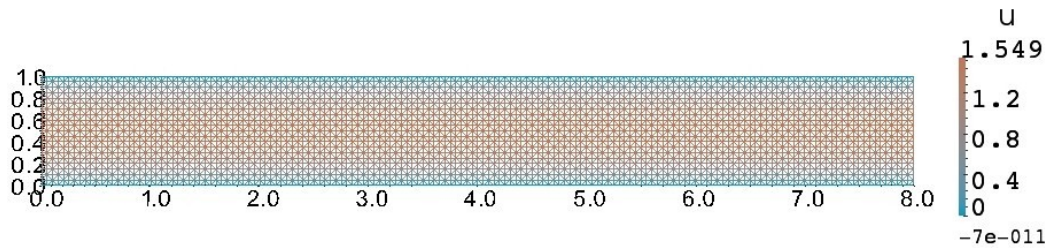

Figure 5.1: Plane Poiseuille Flow simulation domain mesh and horizontal velocity distribution.

Figure 5.1 shows the simulation domain and mesh, and the horizontal velocity distribution. Note that Dirichlet boundary conditions are imposed. The result of the simulation for pressure along the horizontal line passing through the center of the domain is shown in figure 5.2 for Reynolds numbers 10 and 100. These results show that the expected velocity profile and pressure gradient are obtained. Since the velocity profile is parabolic and the pressure gradient is linear the computed solution for this case problem is accurate up to machine precision, for various values of physical parameters, thus corroborating the correctness of the implementation of viscous and pressure terms.
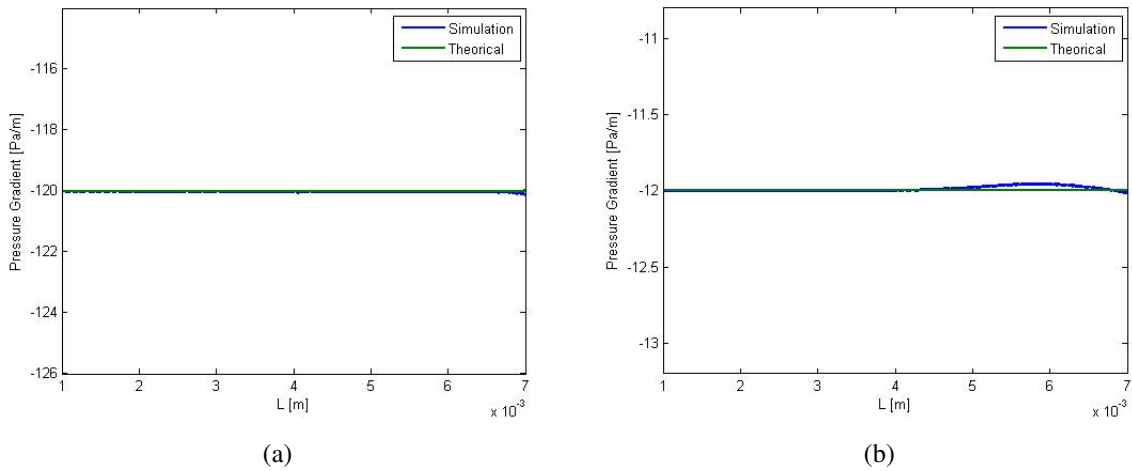
(a)   (b)

Figure 5.2: Pressure gradient evaluation for plane poiseuille flow: (a) with Reynolds number equal to 10, (b) with Reynolds number equal to 100.

5.1.2   Driven cavity flow

The driven cavity flow was simulated in order to validate the implementation of the advection terms. This validation case computes the laminar incompressible flow for a 2D driven cavity at various Reynolds numbers. Particularly, the case of Reynolds number equal to 10.000 is studied in detail. The domain and the boundary conditions employed in the simulations are illustrated in figure 5.3.
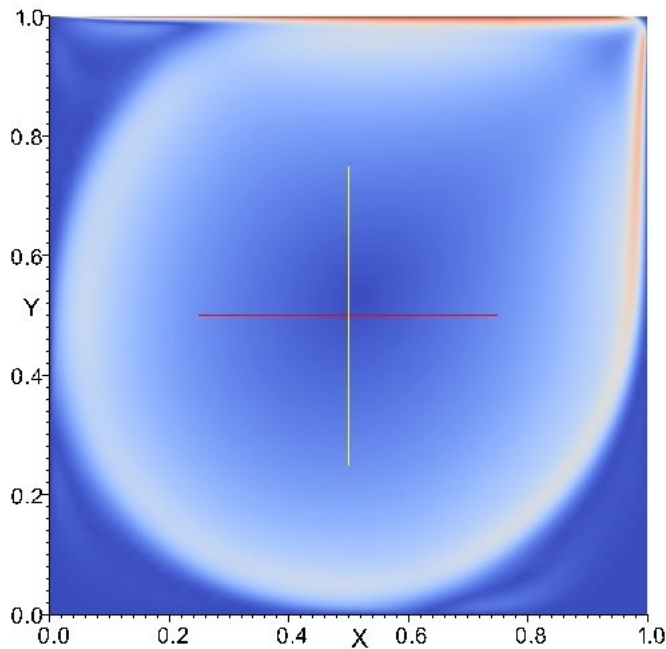

Figure 5.3: Driven cavity flow steady state velocity distribution at Reynolds 10000.

The domain is a $L \times L$ square. Boundary conditions are non-slip (prescribed velocity) at the four walls. The cavity is driven by a translating plate with velocity U at the top of the cavity.

At steady state, the position of the vortices show good agreement with those reported in the literature (HACHEM et al., 2010). Also, as shown in figure 5.5, the velocity profiles along the vertical and horizontal lines passing through the geometric center of the cavity match those described in the literature (ERTURK, 2008), thus validating the implementation of the advection terms in the SUPG implementation. Also, the position and amount of vortices of the computed data match those of the literature as seen in figure 5.4.
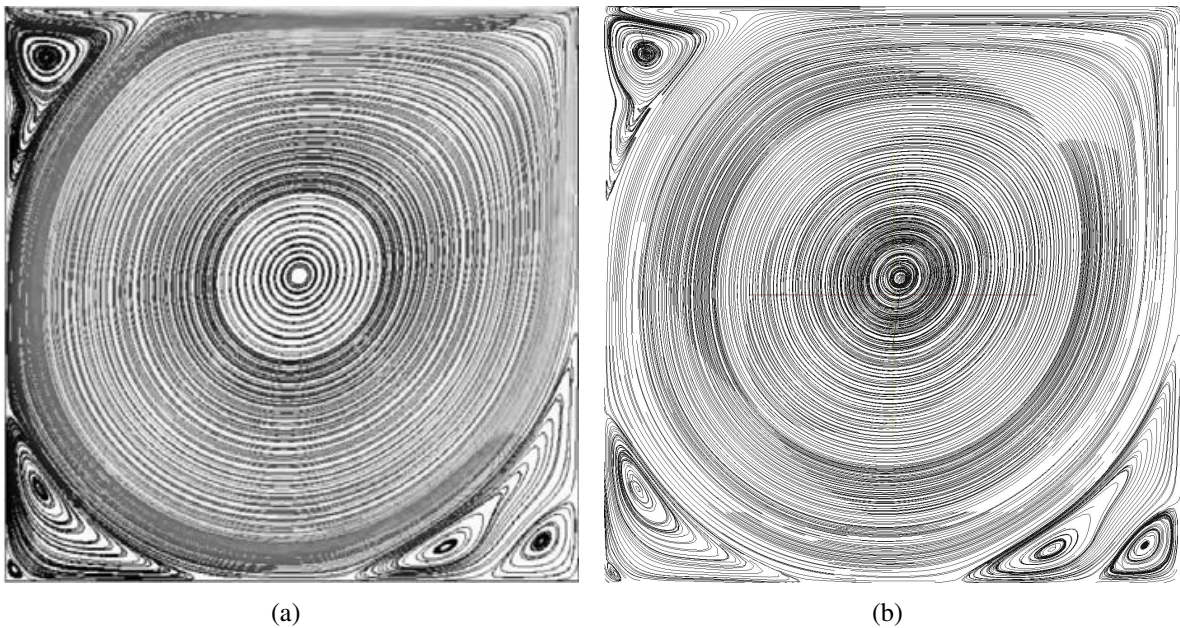

<div align="center">(a)   (b)</div>

Figure 5.4: Comparison between streamlines given by literature and computed data for Re=10.000: (a) (HACHEM et al., 2010), (b) computed data.

5.2 Capillary Pressure Evaluation

We perform this test to validate the computation of the interfacial tension and the capillary pressure. Here, we perform calculations of an spherical fluid bubble or drop of one phase immerse into a fluid of another phase with different grid sizes. These are compared with analytical results obtained by the Young-Laplace (BRACKBILL; KOTHE; ZEMACH, 1992) equation for capillary pressure,

$$\Delta p = \sigma \kappa = \sigma \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \tag{5.2}$$

where Δp is the pressure difference across the fluid interface, σ is the surface tension, and R_1 and R_2 are the principal radii of curvature. Table 5.2 describes the physical parameters used for
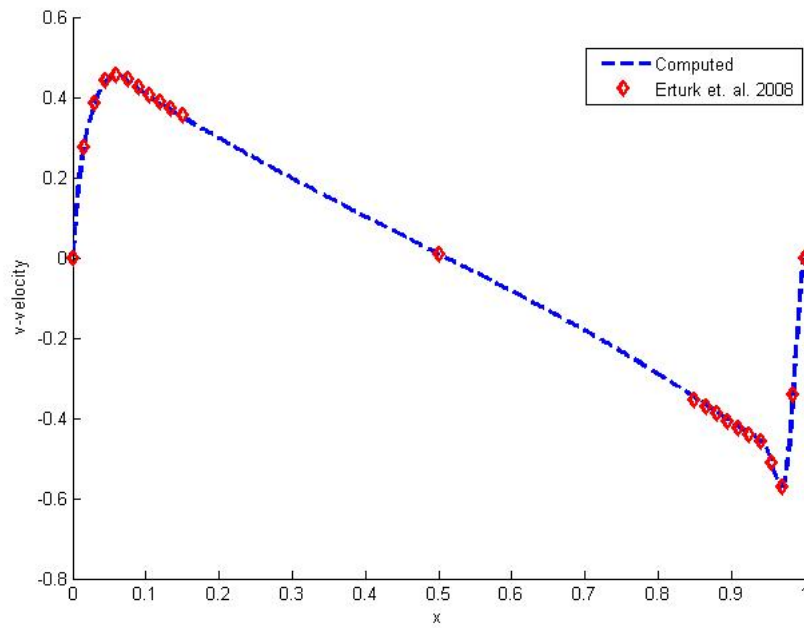
(a)
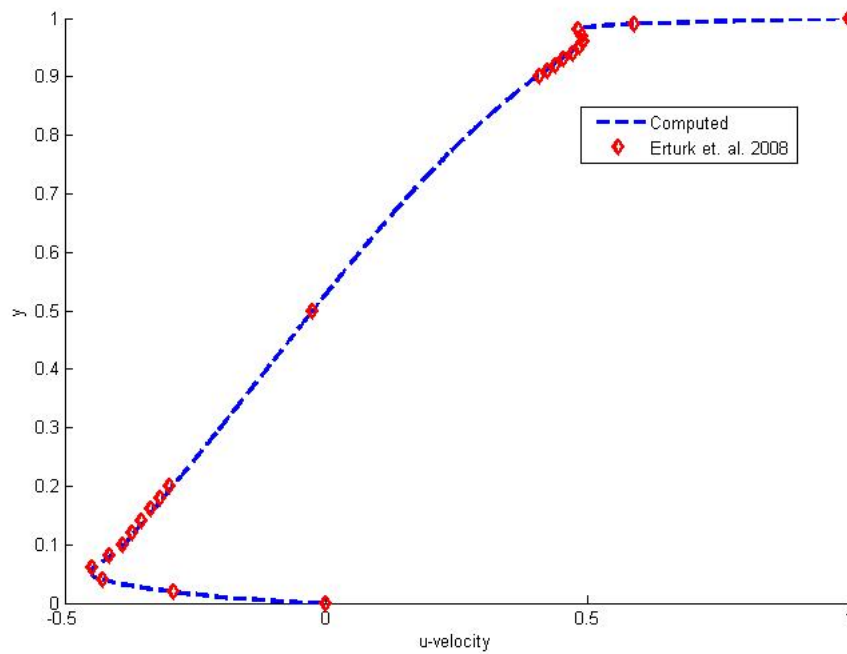
(b)

Figure 5.5: Velocity profiles along the vertical and horizontal lines passing through the geometric center of the cavity: (a) Computed v-velocity profiles along horizontal line passing through the geometric center of the cavity, (b) Computed u-velocity profiles along vertical line passing through the geometric center of the cavity.

the simulations.

Capillary Pressure Evaluation		
Physical Parameters		
Reference density	ρ_∞	1000 kg/m^3
Reference viscosity	μ_∞	$8.9 \cdot 10^{-4}$ Pa·s
Surface tension coefficient	σ	0.02361 N/m
Diameter of the bubble	D	0.04 m
Relative density of external fluid	ρ_1	1
Relative density of internal fluid	ρ_2	0.5
Relative viscosity of external fluid	μ_1	1
Relative viscosity of internal fluid	μ_2	0.05
Non-dimensionalization Parameters		
Reference Length	L	0.04 m
Reference Velocity ($\sqrt{gL}$)	U	0.626 m
Non-dimensional Numbers		
Reynolds Number	Re	10.91
Weber Number	We	1.0
Froude Number*	Fr	–

Table 5.2: Physical parameters used for capillary pressure evaluation. *No gravity field involved $1/Fr = 0$.

Under these conditions, the analytic value of the pressure jump at the interface is deducible from the Young-Laplace equation.

5.3 Frequency of Oscillation

The purpose of this test is to validate the effects of surface tension on an initially elliptical bubble immerse on a continuous flow without any gravity field. For this problem exists an analytical solution for the oscillation frequency. Assuming the viscous effects to be small, and that we are working on an infinite domain, this analytic expression is given by (LAMB, 1945),

$$\omega^2 = \frac{(n^3 - n)\sigma}{(\rho_2 + \rho_1)\rho_\infty R^3} \tag{5.3}$$

With an amplitude decay given by

$$a(t) = a_0 e^{-t/\tau} \tag{5.4}$$

where $\tau = R/5\nu$, ν is the kinematic viscosity, a_0 is the initial amplitude, ω is the angular velocity for the frequency of oscillation. Figure 5.6 shows a comparison between the numerical result and the analytical function given by,

$$y(t) = y_0 + a_0 e^{-t/\tau} \cos(\omega t) \tag{5.5}$$

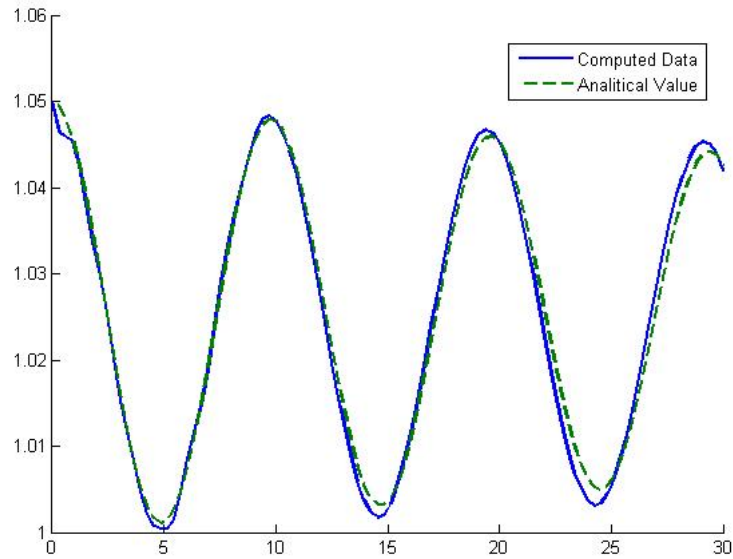The physical parameters used for these simulations are shown in Table 5.3.

Figure 5.6: Oscillation of the bubble diameter as a function of non-dimensional time.

Frequency of Oscillation Evaluation		
Physical Parameters		
Reference density	ρ_∞	1000 kg/m^3
Reference viscosity	μ_∞	$8.9 \cdot 10^{-4}$ Pa·s
Surface tension coefficient	σ	0.02361 N/m
Diameter of the bubble	D	0.04 m
Relative density of external fluid	ρ_1	1
Relative density of internal fluid	ρ_2	100
Relative viscosity of external fluid	μ_1	0.1
Relative viscosity of internal fluid	μ_2	0.35
Non-dimensionalization Parameters		
Reference Length	L	0.04 m
Reference Velocity	U	0.02429 m
Non-dimensional Numbers		
Reynolds Number	Re	1091.91
Weber Number	We	1.0
Froude Number*	Fr	–

Table 5.3: Physical parameters used for frequency of oscillation evaluation. *No gravity field involved $1/Fr = 0$.

5.4 Numerical experiments

In this section, we present the results of the computation of two numerical experiments. The nondimensional parameters that were used to characterize each problem are: The Morton number Mo, the Eötvös number Eo and the Froude number Fr.

5.4.1   Rising bubble

The first experiment consists on an initially stationary bubble that rises due the effect of gravitational forces. The parameters that were used for these simulations are Fr=10.01, Eo=1000 and Mo=0.01. The time step used was variable.

| (a) | (b) | (c) | (d) |
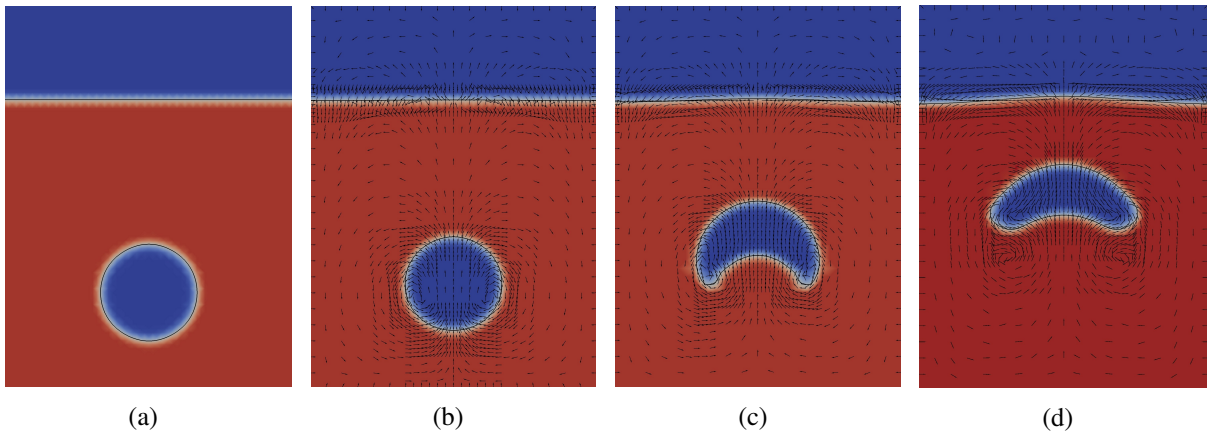
Figure 5.7: Numerical experiment on rising bubble: (a) t=0.0, (b) t=5.0, (c) t=15.0, (d) t=25.1

5.4.2   Bubble Coalescence

The second experiment focuses on bubble coalescence. The parameters that were used for these simulations are $Fr = 0.319$, $Eo = 10$ and $Eo = 0.1$.
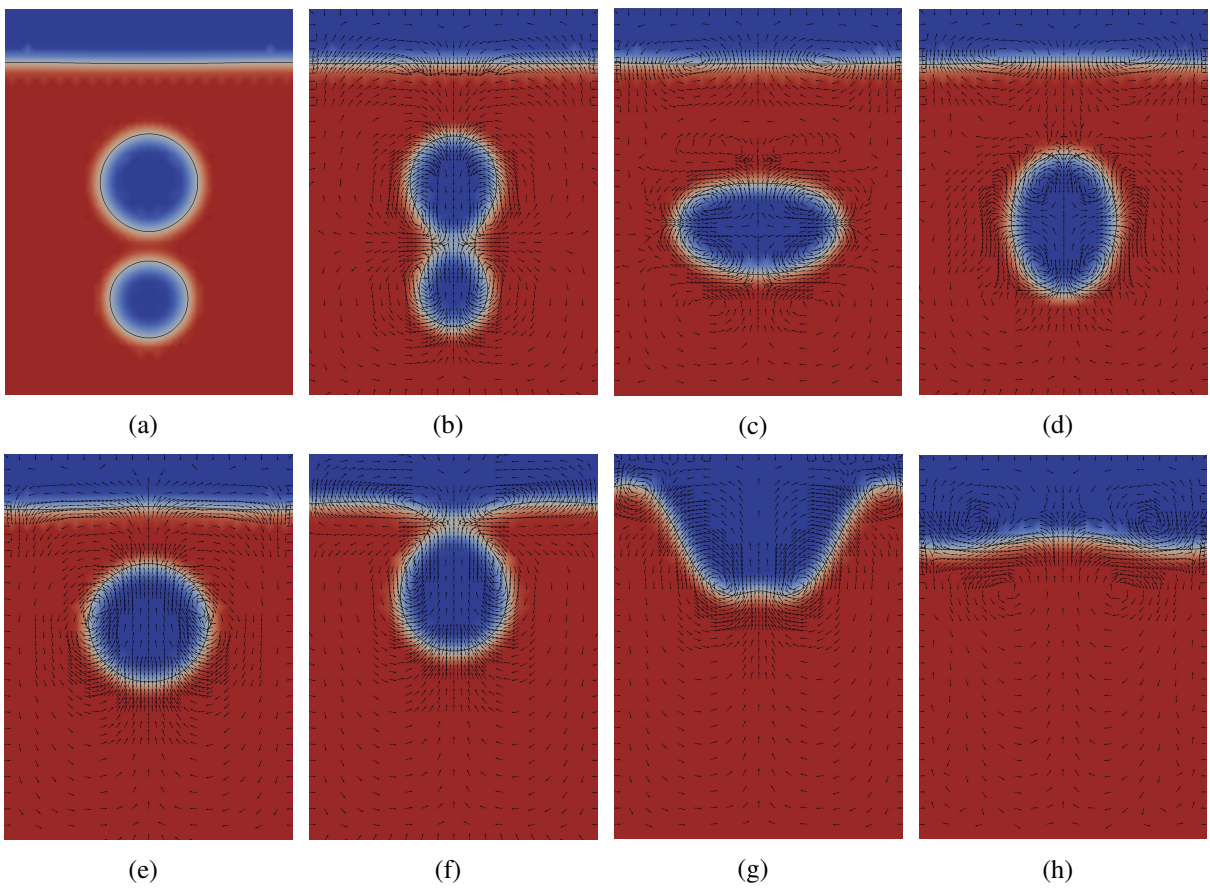
Figure 5.8: Numerical experiment on rising bubble: (a) t=0.0, (b) t=0.077, (c) t=0.179, (d) t=0.262, (e) t=0.707, (f) t=0.937, (g) t=1.030, (h) t=1.150

CHAPTER 6

SUMMARY

The method presented in this paper is an adaptive finite element level set approach that deals with the mass conservation problems that are inherent to the level set method, through mesh refinement capabilities and accurate correction techniques for keeping the level set function a sign function. This method results on effectively conserving mass properties while easily dealing with topological changes of the fronts, such as bubble coalescence and changes on free surface. The code implements a finite element unstructured mesh as well as a control procedure using *libmesh*, an adaptive C++ finite element library for simulating partial differential equations.

6.1 Future Works

We can see from the last simulation that the bubbles start the merging process when they are close enough so that the smeared out surface tension force of one bubble cancels with the one of the other, reducing the surface tension in that region. In reality, this should happen when the region that defines one bubble intersects the region defined by the other bubble. From equation (2.61) we can see that the surface tension is defined by the smeared out derivative of the surface tension, thus accuracy can be increased by reducing the numerical smearing of the interface. However, by doing this we are introducing numerical instabilities, thus, the region is refined and the simulation becomes highly computationally intensive. Now, with the difficulties of the method identified, we can think on new methods to achieve even more accurate results on coarse meshes, such as the ghost fluid method shown in (OSHER; FEDKIW, 2003) to directly define the pressure jump across the interface.

A.1 Generalized Navier-Stokes element matrices and vectors

As we have seen, we can split the element matrix $\mathbf{K^e}$ and element vector $\mathbf{F^e}$ into submatrices each corresponding to the unknown nodal values of the velocity components u and v (assuming $n_{sd} = 2$) and pressure p,

$$
K^e = \begin{bmatrix} K_{uu}^e & K_{uv}^e & K_{up}^e \\ K_{vu}^e & K_{vv}^e & K_{vp}^e \\ K_{pu}^e & K_{pv}^e & K_{pp}^e \end{bmatrix} \quad F^e = \begin{bmatrix} F_u^e \\ F_v^e \\ F_p^e \end{bmatrix} \tag{A.1}
$$

Then, starting from equation (4.7) and assuming $\tau_{\mathrm{PSPG}} = \tau_{\mathrm{LSIC}} = 0$, we can calculate each element submatrix writing the terms in the same order, as follows,

$$
\begin{aligned}
Fu_a^e = \int_\Omega (\\
+ \left(\rho_r u_n N_a \right) / \left(\Delta t \right) \\
+ \left(\left(2\frac{\partial u_n}{\partial x}\left(\frac{\partial N_a}{\partial x}\right) + \left(\frac{\partial u_n}{\partial y} + \frac{\partial v_n}{\partial x}\right)\left(\frac{\partial N_a}{\partial y}\right)\right) \mu_r\left(\theta - 1\right)\right)/Re \\
+ - \left(\rho_r \left((\alpha - 1)\frac{\partial u_n}{\partial x}\left(\theta - 1\right) u_n - \right. \right. \\
\left. \left. \alpha\theta\left(\frac{\partial u_{n+1}^i}{\partial x}u + \frac{\partial u_{n+1}^i}{\partial y}v\right) + (\alpha - 1)\frac{\partial u_n}{\partial y}\left(\theta - 1\right) v_n \right) N_a \right) \\
+ 0
\end{aligned}
$$

$$+ \frac{k}{We} \frac{\partial H}{\partial x} N_a$$

$$+ - \left(\left(\rho_r \tau_{\text{SUPG}} u_n \left(\left(\frac{\partial N_a}{\partial x} \right) \left(- (\alpha u) + (\alpha - 1) u_n \right) + \right. \right. \right.$$

$$\left. \left. \left. \left(\frac{\partial N_a}{\partial y} \right) \left(- (\alpha v) + (\alpha - 1) v_n \right) \right) \right) / (\Delta t) \right)$$

$$+ \left(\left(\left(\frac{\partial^2 u_n}{\partial x^2} + \frac{\partial^2 u_n}{\partial y^2} \right) \mu_r (\theta - 1) \tau_{\text{SUPG}} \right. \right.$$

$$\left(\left(\frac{\partial N_a}{\partial x} \right) \left(- (\alpha u) + (\alpha - 1) u_n \right) + \right.$$

$$\left. \left. \left. \left(\frac{\partial N_a}{\partial y} \right) \left(- (\alpha v) + (\alpha - 1) v_n \right) \right) \right) / Re \right)$$

$$+ \rho_r \tau_{\text{SUPG}} \left((\alpha - 1) \frac{\partial u_n}{\partial x} (\theta - 1) u_n - \right.$$

$$\alpha \theta \left(\frac{\partial u_{n+1}^i}{\partial x} u + \frac{\partial u_{n+1}^i}{\partial y} v \right) + (\alpha - 1) \frac{\partial u_n}{\partial y} (\theta - 1) v_n \right)$$

$$\left(\left(\frac{\partial N_a}{\partial x} \right) \left(- (\alpha u) + (\alpha - 1) u_n \right) + \right.$$

$$\left. \left(\frac{\partial N_a}{\partial y} \right) \left(- (\alpha v) + (\alpha - 1) v_n \right) \right)$$

$$+ - \left(\frac{k}{We} \frac{\partial H}{\partial x} \tau_{\text{SUPG}} \left(\left(\frac{\partial N_a}{\partial x} \right) \left(- (\alpha u) + (\alpha - 1) u_n \right) + \right. \right.$$

$$\left. \left. \left(\frac{\partial N_a}{\partial y} \right) \left(- (\alpha v) + (\alpha - 1) v_n \right) \right) \right)$$

$$\left. \right) d\Omega$$

$$F v_a^e = \int_\Omega \big($$

$$+ \left(\rho_r v_n N_a \right) / (\Delta t)$$

$$+ \left(\left(\left(\frac{\partial u_n}{\partial y} + \frac{\partial v_n}{\partial x} \right) \left(\frac{\partial N_a}{\partial x} \right) + 2 \frac{\partial v_n}{\partial y} \left(\frac{\partial N_a}{\partial y} \right) \right) \mu_r (\theta - 1) \right) / Re$$

$$+ - \left(\rho_r \left((\alpha - 1) \frac{\partial v_n}{\partial x} (\theta - 1) u_n - \right. \right.$$

$$\left. \left. \alpha \theta \left(\frac{\partial v_{n+1}^i}{\partial x} u + \frac{\partial v_{n+1}^i}{\partial y} v \right) + (\alpha - 1) \frac{\partial v_n}{\partial y} (\theta - 1) v_n \right) N_a \right)$$

$$+ 0$$

$$+ \left(\frac{k}{We} \frac{\partial H}{\partial y} + \frac{\rho_r}{Fr^2} \right) N_a$$

$$+ - \left(\left(\rho_r \tau_{\text{SUPG}} v_n \left(\left(\frac{\partial N_a}{\partial x} \right) \left(- (\alpha u) + (\alpha - 1) u_n \right) + \right. \right. \right.$$

$$\left(\frac{\partial N_a}{\partial y}\right)\left(-\left(\alpha v\right)+\left(\alpha-1\right)v_n\right)\right)\right)/\left(\Delta t\right)\right)$$

$$+\left(\left(\left(\frac{\partial^2 v_n}{\partial x^2}+\frac{\partial^2 v_n}{\partial y^2}\right)\mu_r\left(\theta-1\right)\tau_{\text{SUPG}}\right.$$

$$\left(\left(\frac{\partial N_a}{\partial x}\right)\left(-\left(\alpha u\right)+\left(\alpha-1\right)u_n\right)+\right.$$

$$\left.\left(\frac{\partial N_a}{\partial y}\right)\left(-\left(\alpha v\right)+\left(\alpha-1\right)v_n\right)\right)\right)/Re$$

$$+\rho_r\tau_{\text{SUPG}}\left(\left(\alpha-1\right)\frac{\partial v_n}{\partial x}\left(\theta-1\right)u_n-\right.$$

$$\alpha\theta\left(\frac{\partial v_{n+1}^i}{\partial x}u+\frac{\partial v_{n+1}^i}{\partial y}v\right)+\left(\alpha-1\right)\frac{\partial v_n}{\partial y}\left(\theta-1\right)v_n\right)$$

$$\left(\left(\frac{\partial N_a}{\partial x}\right)\left(-\left(\alpha u\right)+\left(\alpha-1\right)u_n\right)+\right.$$

$$\left.\left(\frac{\partial N_a}{\partial y}\right)\left(-\left(\alpha v\right)+\left(\alpha-1\right)v_n\right)\right)$$

$$+-\left(\left(\frac{k}{We}\frac{\partial H}{\partial y}+\frac{\rho_r}{Fr^2}\right)\tau_{\text{SUPG}}\left(\left(\frac{\partial N_a}{\partial x}\right)\left(-\left(\alpha u\right)+\right.\right.\right.$$

$$\left.\left.\left(\alpha-1\right)u_n\right)+\left(\frac{\partial N_a}{\partial y}\right)\left(-\left(\alpha v\right)+\left(\alpha-1\right)v_n\right)\right)\right)$$

$$\left.\right)d\Omega$$

$$Kuu_{ab}^e=\int_\Omega(\,$$

$$+\left(\rho_r N_b N_a\right)/\left(\Delta t\right)$$

$$+\left(\left(\left(2\left(\frac{\partial N_b}{\partial x}\right)\left(\frac{\partial N_a}{\partial x}\right)+\left(\frac{\partial N_b}{\partial y}\right)\left(\frac{\partial N_a}{\partial y}\right)\right)\mu_r\theta\right)/Re\right.$$

$$+\rho_r\left(-\left(\alpha\frac{\partial u_n}{\partial x}\left(\theta-1\right)N_b\right)+\right.$$

$$\theta\left(\alpha\frac{\partial u_{n+1}^i}{\partial x}N_b+\left(\frac{\partial N_b}{\partial x}\right)\left(\alpha u+u_n-\alpha u_n\right)+\right.$$

$$\left.\left.\left(\frac{\partial N_b}{\partial y}\right)\left(\alpha v+v_n-\alpha v_n\right)\right)\right)N_a$$

$$+0$$

$$+0$$

$$+\left(\rho_r\tau_{\text{SUPG}}N_b\left(\left(\frac{\partial N_a}{\partial x}\right)\left(\alpha u+u_n-\alpha u_n\right)+\right.\right.$$

$$\left.\left.\left(\frac{\partial N_a}{\partial y}\right)\left(\alpha v+v_n-\alpha v_n\right)\right)\right)/\left(\Delta t\right)$$

$$+ \left(\left(\left(\frac{\partial^2 N_b}{\partial x^2} + \frac{\partial^2 N_b}{\partial y^2} \right) \mu_r \theta \tau_{\text{SUPG}} \right. \right.$$

$$\left(\left(\frac{\partial N_a}{\partial x} \right) \left(-\left(\alpha u \right) + \left(\alpha - 1 \right) u_n \right) + \right.$$

$$\left. \left. \left(\frac{\partial N_a}{\partial y} \right) \left(-\left(\alpha v \right) + \left(\alpha - 1 \right) v_n \right) \right) \right) / Re$$

$$+ \rho_r \tau_{\text{SUPG}} \left(\left(\left(\frac{\partial N_a}{\partial x} \right) \left(-\left(\alpha u \right) + \left(\alpha - 1 \right) u_n \right) + \right. \right.$$

$$\left. \left(\frac{\partial N_a}{\partial y} \right) \left(-\left(\alpha v \right) + \left(\alpha - 1 \right) v_n \right) \right) \left(\alpha \frac{\partial u_n}{\partial x} \left(\theta - 1 \right) N_b - \right.$$

$$\alpha \theta \left(\frac{\partial u_{n+1}^i}{\partial x} N_b + \left(\frac{\partial N_b}{\partial x} \right) \left(u - u_n \right) + \left(\frac{\partial N_b}{\partial y} \right) \left(v - v_n \right) \right) -$$

$$\left. \theta \left(\left(\frac{\partial N_b}{\partial x} \right) u_n + \left(\frac{\partial N_b}{\partial y} \right) v_n \right) \right)$$

$$+ 0$$

$$\left. \right) d\Omega$$

$$Kuv_{ab}^e = \int_\Omega ($$

$$+ 0$$

$$+ \left(\left(\frac{\partial N_b}{\partial x} \right) \left(\frac{\partial N_a}{\partial y} \right) \mu_r \theta \right) / Re$$

$$+ \alpha \rho_r \left(\frac{\partial u_n}{\partial y} + \frac{\partial u_{n+1}^i}{\partial y} \theta - \frac{\partial u_n}{\partial y} \theta \right) N_b N_a$$

$$+ 0$$

$$+ 0$$

$$+ 0$$

$$+ 0$$

$$+ \alpha \rho_r \left(\frac{\partial u_n}{\partial y} \left(\theta - 1 \right) - \frac{\partial u_{n+1}^i}{\partial y} \theta \right) \tau_{\text{SUPG}} N_b$$

$$\left(\left(\frac{\partial N_a}{\partial x} \right) \left(-\left(\alpha u \right) + \left(\alpha - 1 \right) u_n \right) + \right.$$

$$\left. \left(\frac{\partial N_a}{\partial y} \right) \left(-\left(\alpha v \right) + \left(\alpha - 1 \right) v_n \right) \right)$$

$$+ 0$$

$$\left. \right) d\Omega$$

$$Kvu^e_{ab} = \int_\Omega \big($$

$$+0$$

$$+\left(\left(\frac{\partial N_b}{\partial y}\right)\left(\frac{\partial N_a}{\partial x}\right)\mu_r\theta\right)/Re$$

$$+\alpha\rho_r\left(\frac{\partial v_n}{\partial x} + \frac{\partial v^i_{n+1}}{\partial x}\theta - \frac{\partial v_n}{\partial x}\theta\right)N_b N_a$$

$$+0$$

$$+0$$

$$+0$$

$$+0$$

$$+\alpha\rho_r\left(\frac{\partial v_n}{\partial x}(\theta-1) - \frac{\partial v^i_{n+1}}{\partial x}\theta\right)\tau_{\text{SUPG}}N_b$$

$$\left(\left(\frac{\partial N_a}{\partial x}\right)(-(\alpha u) + (\alpha-1)u_n) + \right.$$

$$\left. \left(\frac{\partial N_a}{\partial y}\right)(-(\alpha v) + (\alpha-1)v_n)\right)$$

$$+0$$

$$\big)\,d\Omega$$

$$Kvv^e_{ab} = \int_\Omega \big($$

$$+ (\rho_r N_b N_a)/(\Delta t)$$

$$+\left(\left(\left(\frac{\partial N_b}{\partial x}\right)\left(\frac{\partial N_a}{\partial x}\right) + 2\left(\frac{\partial N_b}{\partial y}\right)\left(\frac{\partial N_a}{\partial y}\right)\right)\mu_r\theta\right)/Re$$

$$+\rho_r\left(-\left(\alpha\frac{\partial v_n}{\partial y}(\theta-1)N_b\right) + \right.$$

$$\theta\left(\left(\frac{\partial N_b}{\partial x}\right)u_n + \left(\frac{\partial N_b}{\partial y}\right)v_n + \alpha\left(\left(\frac{\partial N_b}{\partial x}\right)(u-u_n) + \left(\frac{\partial N_b}{\partial y}\right)v + \right.\right.$$

$$\left.\left.\left.\frac{\partial v^i_{n+1}}{\partial y}N_b - \left(\frac{\partial N_b}{\partial y}\right)v_n\right)\right)\right)N_a$$

$$+0$$

$$+0$$

$$+\left(\rho_r\tau_{\text{SUPG}}N_b\left(\left(\frac{\partial N_a}{\partial x}\right)(\alpha u + u_n - \alpha u_n) + \right.\right.$$

$$\left.\left.\left(\frac{\partial N_a}{\partial y}\right)(\alpha v + v_n - \alpha v_n)\right)\right)/(\Delta t)$$

$$+ \left(\left(\left(\frac{\partial^2 N_b}{\partial x^2} + \frac{\partial^2 N_b}{\partial y^2} \right) \mu_r \theta \tau_{\text{SUPG}} \right. \right.$$

$$\left(\left(\frac{\partial N_a}{\partial x} \right) \left(- \left(\alpha u \right) + \left(\alpha - 1 \right) u_n \right) + \right.$$

$$\left. \left. \left(\frac{\partial N_a}{\partial y} \right) \left(- \left(\alpha v \right) + \left(\alpha - 1 \right) v_n \right) \right) \right) / Re$$

$$+ \rho_r \tau_{\text{SUPG}} \left(\left(\frac{\partial N_a}{\partial x} \right) \left(- \left(\alpha u \right) + \left(\alpha - 1 \right) u_n \right) + \right.$$

$$\left(\frac{\partial N_a}{\partial y} \right) \left(- \left(\alpha v \right) + \left(\alpha - 1 \right) v_n \right) \right) \left(\alpha \frac{\partial v_n}{\partial y} \left(\theta - 1 \right) N_b - \right.$$

$$\alpha \theta \left(\left(\frac{\partial N_b}{\partial x} \right) \left(u - u_n \right) + \frac{\partial v_{n+1}^i}{\partial y} N_b + \left(\frac{\partial N_b}{\partial y} \right) \left(v - v_n \right) \right) -$$

$$\left. \theta \left(\left(\frac{\partial N_b}{\partial x} \right) u_n + \left(\frac{\partial N_b}{\partial y} \right) v_n \right) \right)$$

$$+ 0$$

$$\bigg) \, d\Omega$$

$$Kup_{ab}^e = \int_\Omega \bigg($$

$$+ 0$$

$$+ - \left(\left(\frac{\partial N_a}{\partial x} \right) M_b \right)$$

$$+ 0$$

$$+ 0$$

$$+ 0$$

$$+ 0$$

$$+ \left(\frac{\partial M_b}{\partial x} \right) \tau_{\text{SUPG}} \left(\left(\frac{\partial N_a}{\partial x} \right) \left(\alpha u + u_n - \alpha u_n \right) + \right.$$

$$\left. \left(\frac{\partial N_a}{\partial y} \right) \left(\alpha v + v_n - \alpha v_n \right) \right)$$

$$+ 0$$

$$+ 0$$

$$\bigg) \, d\Omega$$

$$Kvp_{ab}^e = \int_\Omega \bigg($$

$$+0$$

$$+ - \left(\left(\frac{\partial N_a}{\partial y} \right) M_b \right)$$

$$+0$$

$$+0$$

$$+0$$

$$+0$$

$$+ \left(\frac{\partial M_b}{\partial y} \right) \tau_{\text{SUPG}} \left(\left(\frac{\partial N_a}{\partial x} \right) (\alpha u + u_n - \alpha u_n) + \right.$$
$$\left. \left(\frac{\partial N_a}{\partial y} \right) (\alpha v + v_n - \alpha v_n) \right)$$

$$+0$$

$$+0$$

$$\Big) d\Omega$$

$$Kpu_{ab}^e = \int_\Omega ($$

$$+0$$

$$+0$$

$$+0$$

$$+ \left(\frac{\partial N_b}{\partial x} \right) \theta M_a$$

$$+0$$

$$+0$$

$$+0$$

$$+0$$

$$+0$$

$$\Big) d\Omega$$

$$Kpv_{ab}^e = \int_\Omega ($$

$$+0$$

$$+0$$

$$+0$$
$$+\left(\frac{\partial N_b}{\partial y}\right)\theta M_a$$
$$+0$$
$$+0$$
$$+0$$
$$+0$$
$$+0$$
$$\Bigg)\,d\Omega$$

$$Kpp^e_{ab}=0$$

REFERENCES

BRACKBILL, J.; KOTHE, D.; ZEMACH, C. A continuum method for modeling surface tension. *Journal of computational physics*, Elsevier, v. 100, n. 2, p. 335–354, 1992.

BROOKS, A.; HUGHES, T. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer methods in applied mechanics and engineering*, Elsevier, v. 32, n. 1-3, p. 199–259, 1982.

DONEA, J.; HUERTA, A. Finite Element Methods for Flow Probelms. Wiley Online Library, 2003.

ERTURK, E. Numerical solutions of 2-D steady incompressible flow over a backward-facing step, Part I: High Reynolds number solutions. *Computers & Fluids*, Elsevier, v. 37, n. 6, p. 633–655, 2008. ISSN 0045-7930.

ESMAEELI, A.; TRYGGVASON, G. Direct numerical simulations of bubbly flows. Part 1. Low Reynolds number arrays. *Journal of Fluid Mechanics*, Cambridge University Press, v. 377, p. 313–345, 1998.

GROSS, S.; REICHELT, V.; REUSKEN, A. A finite element based level set method for two-phase incompressible flows. *Computing and Visualization in Science*, Springer, v. 9, n. 4, p. 239–257, 2006.

HACHEM, E. et al. Stabilized finite element method for incompressible flows with high Reynolds number. *Journal of Computational Physics*, Elsevier, 2010. ISSN 0021-9991.

HAUKE, G. *An introduction to fluid mechanics and transport phenomena*. [S.l.]: Springer Verlag, 2008. ISBN 1402085362.

HEINRICH, J.; PEPPER, D. *Intermediate finite element method: fluid flow and heat transfer applications*. [S.l.]: Taylor & Francis Group, 1999.

HUGHES, T.; MALLET, M. A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems* 1. *Computer Methods in Applied Mechanics and Engineering*, Elsevier, v. 58, n. 3, p. 305–328, 1986.

KELLY, D. W. et al. A posteriori error analysis and adaptive processes in the finite element method: Part I Error analysis. *Int. J. Num. Meth. Engng.*, v. 19, p. 1593–1619, 1983.

KIRK, B. Adaptive finite element simulation of flow and transport applications on parallel computers. D., THE UNIVERSITY OF TEXAS AT AUSTIN, 2007.

KIRK, B. et al. `libMesh`: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations. *Engineering with Computers*, v. 22, n. 3–4, p. 237–254, 2006.

LAMB, H. *Hydrodynamics.* [S.l.]: Dover Publications, New York, 1945.

OSHER, S.; FEDKIW, R. *Level set methods and dynamic implicit surfaces*. [S.l.]: Springer Verlag, 2003.

OSHER, S.; SETHIAN, J. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of computational physics*, Elsevier, v. 79, n. 1, p. 12–49, 1988.

PENG, D. et al. A PDE-Based Fast Local Level Set Method. *Journal of Computational Physics*, Elsevier, v. 155, n. 2, p. 410–438, 1999.

RABELLO, G. et al. An ALE finite element method for the simulation of 3D multiphase flows. *12th Brazilian Congress of Thermal Engineering and Sciences*, 2008.

SOUSA, F.; MANGIAVACCHI, N. A Lagrangian level-set approach for the simulation of incompressible two-fluid flows. *International Journal for Numerical Methods in Fluids*, John Wiley & Sons, v. 47, n. 10-11, p. 1393–1401, 2005.

SOUSA, F. D. et al. A front-tracking/front-capturing method for the simulation of 3D multi-fluid flows with free surfaces. *Journal of Computational Physics*, Elsevier, v. 198, n. 2, p. 469–499, 2004.

SUSSMAN, M.; FATEMI, E. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM Journal on Scientific Computing*, Citeseer, v. 20, n. 4, p. 1165–1191, 1999.

SUSSMAN, M.; SMEREKA, P.; OSHER, S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of computational Physics*, New York, Academic Press., v. 114, n. 1, p. 146–159, 1994.

TEZDUYAR, T.; OSAWA, Y. Finite element stabilization parameters computed from element matrices and vectors. *Comput. Methods Appl. Mech. Engrg*, v. 190, n. 3-4, p. 411–430, 2000.

TORNBERG, A.; ENGQUIST, B. A finite element based level-set method for multiphase flow applications. *Computing and Visualization in Science*, Springer, v. 3, n. 1, p. 93–101, 2000.