Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

Leandro Marques dos Santos

An ALE-FE Method for Blood Flow Dynamics in Coronary Arteries Using the Vorticity-Streamfunction Formulation

Rio de Janeiro, Brazil

2020

Leandro Marques dos Santos

An ALE-FE Method for Blood Flow Dynamics in Coronary Arteries Using the Vorticity-Streamfunction Formulation

Master's Thesis presented to the Mechanical Engineering Graduate Program of State University of Rio de Janeiro (UERJ) as a pratial requirement to obtain the degree of Master in Sciences. Field of concentration: Transport Phenomena.

Advisor: Prof. Gustavo Rabello dos Anjos, Ph.D.

Co-Advisor: Prof. José da Rocha Miranda Pontes, D.Sc.

Rio de Janeiro, Brazil

2020

_____   _____
              Assinatura   Data

Leandro Marques dos Santos

An ALE-FE Method for Vorticity-Streamfunction Formulation with Species Transport Equation

Master's Thesis presented to the Mechanical Engineering Graduate Program of State University of Rio de Janeiro (UERJ) as a partial requirement to obtain the degree of Master in Sciences. Field of concentration: Transport Phenomena.

Approved on: August 14, 2020

Examining Committee:

Prof. Gustavo Rabello dos Anjos, Ph.D. (Advisor)
Federal University of Rio de Janeiro - (COPPE/UFRJ)

Prof. José da Rocha Miranda Pontes, D.Sc. (Co-Advisor)
State University of Rio de Janeiro - (PPGEM/UERJ)

Prof. Norberto Mangiavacchi, Ph.D.
State University of Rio de Janeiro - (PPGEM/UERJ)

Rachel Manhães de Lucena, D.Sc.
State University of Rio de Janeiro - (PPGEM/UERJ)

Prof. Gustavo Charles Peixoto de Oliveira, D.Sc.
Federal University of Paraíba - (PPGEM/UFPB)

Rio de Janeiro, Brazil

2020

ACKNOWLEDGMENTS

ABSTRACT

Marques, Leandro *An ALE-FE Method for Blood Flow Dynamics in Coronary Arteries Using the Vorticity-Streamfunction Formulation.* xxf. Master's Thesis (Master of Mechanical Engineering) - Engineering Departament, State University of Rio de Janeiro (UERJ), Rio de Janeiro, Brazil, 2020.

      The present work aims at developing a computational framework to simulate blood flow in coronary artery with drug-eluting stent placed for several Schmidt number using the Vorticity-Streamfunction Formulation in an Arbitrary Lagrangian-Eulerian (ALE) approach. The blood was modeled as single-phase, incompressible and newtonian fluid. The Navier-Stokes equation is shown according to the vorticity-streamfunction formulation with species transport equation. The Finite Element Method (FEM) is used to solve the governing equations where the Galerkin formulation was used to discretized the governing equations in spatial domain and the semi-Lagrangian method was used to discretized the material derivative using first order backward difference scheme. Moreover, the mesh update procedure performs nodes regularisation allowing to simulate moving boundary problems and the linear systems was solved using Conjugate Gradient Iterative Method. Therefore, the new formulation proposed in this work is suitable to simulate flows in coronary arteries with good accuracy.

Keywords: Abritary Lagrangian-Eulerian; Vorticity-Streamfunction Formulation; Finite Element Method; semi-Lagrangian Method; Drug-Eluting Stent.

RESUMO

Marques, Leandro *Um Método FE-ALE para a Dinâmica do Escoamento Sanguíneo na Artéria Coronária Usando a Formulação Corrente-Vorticidade*. xxf. Dissertação de Mestrado (Mestrado em Engenharia Mecânica) - Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, Brasil, 2020.

O presente trabalho tem como objetivo o desenvolvimento de uma estrutura computacional para simular o escoamento sanguíneo em uma artéria coronária com stent farmacológico implantado para diversos números de Schmidt usando a Formulação Corrente-Vorticidade em uma abordagem Lagrangeana-Euleriana Arbitrária (ALE). O fluido sanguíneo foi modelado como um fluido monofásico, incompressível e newtoniano. A equação de Navier-Stokes é apresentada segundo a formulação corrente-vorticidade com a equação de transporte de espécie química. O Método dos Elementos Finitos (FEM) é usado para resolver as equações de governo, onde a formulação de Galerkin foi usada para discretização no espaço, enquanto o Método semi-Lagrangeano foi usado para discretizar a derivada material usando o *backward difference scheme*. Além disso, o procedimento de remalhamento realiza a regularização dos nós possibilitando simular problemas com contornos móveis e os sistemas lineares foram resolvidos utilizando o Método Iterativo Gradientes Conjugados. Portanto, a nova formulação proposta nesse trabalho é apropriada para simular escoamentos em artéria coronária com boa precisão.

Palavras-chave: Lagrangeana-Euleriana Arbitrária; Formulação Corrente-Vorticidade; Método dos Elementos Finitos; Método semi-Lagrangeano; Stent Farmacológico.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

INTRODUCTION

According to the World Health Organization (WHO), more people die each year from cardiovascular disease (CVD) than any other cause in the world in the last 15 years [?]. It is estimated that 15.2 million people died from CVD in 2016, representing 26.7% of all deaths in the world. In Brazil, about 38% of deaths due to CVD is in the procutive age range (18 to 65 years) and the estimated costs of CVD were R$ 37.1 billion in 2015, that is, 0.7% GDP [?]. About 60% of CVD deaths occurred due to coronary artery disease (CAD). The main cause of CAD is the atherosclerosis which consists of the accumulation of fatty plaques inside the artery wall causing a decrease in lumen diameter. The Atherosclerosis can be prevented with a change in harmful habits such as: cigarette smoking, physical inactivity/low fitness and poor dietary habits [?]. For a corrective approach, however, two treatments can be performed: the *Coronary Artery Bypass Sugery* (CABG) and *Percutaneous Transluminal Coronary Angioplasty* (PTCA). The PTCA a procedure minimally invasive where a wire tube, called *stents* is placed [?].

This work is a continuation of that developed by Marques (2018) [?], however the goals of the current work plan are the development of a Finite Element code using the Lagragian-Eulerian Arbitrary (ALE) description for the Vorticity-Streamfunction Formulation with Species Transport Equation, to implement the semi-Lagrangian Method in place of the Taylor-Galerkin Method and to know the dynamics of blood flow in a coronary artery with atherosclerosis and drug-eluting stent placed for several Schmidt numbers.

The equations that govern the dynamics of blood flow in a coronary artery were developed according to the continuum hypothesis. In this way, the principles of mass conservation, linear momentum conservation and chemical species were used. The blood was considered as an incompressible, newtonian and one-phase fluid, as well as the mass diffusivity was approximated as constant. The Navier-Stokes equation is shown according to the vorticity-streamfunction formulation with species transport equation without internal source term in an Arbitrary Lagrangian-Eulerian (ALE).

The domain were discretized using an unstructured triangular mesh generated by GMSH open source [?] and the governing equations were discretized in spatial domain by the Galerkin Method. Due to decoupling between the velocity and pressure fields provided by vorticity-streamfunction formulation, the linear triangular element can be used without breaking the Babuska-Brezzi restriction [?] [?]. The Navier-Stokes and Species Transport

equations were discretized in time using the semi-Lagragian Method [?] where the searching procedure is discussed.

The computational development was done in Python [?] using Object-Oriented Paradigm (OOP) and the Chapter 4 presents the computational cost for each simulator process in addition to the solution algorithm used and the Laplacian Smoothing Method is discussed. The numerical code validation was performed by comparison between numerical and analytical solutions in three benchmark problems: *Couette Flow, Poiseuille Flow* and *Half Poiseuille Flow*. Moreover, the horizontal and vertical velocities in *Lid-Driven Cavity* was compared with those presented by *Ghia et al.* [?] and *Marchi et al.* [?]. Then, the comparison between the semi-Lagrangian Method for an unstructured linear triangular and a quadratic mesh is presented in a pure convective flow.

The blood flow and the concentration transport in coronary artery were simulated in 2 geometries types as suggested by *Wang et al.* [?], however with some modications to the cartesian coordinates. These geometries types consist of one: (1) coronary artery with atherosclerosis with 40% lumen obstruction and with drug-eluting stent placed; (2) real coronary artery with atherosclerosis and with drug-eluting stent placed. The numerical simulation visualization was performed using *Paraview* open source as proposed by *Henderson (2007)* [?]. This work was organized as follows:

- Introduction
- Chapter 1: Literature Review
- Chapter 2: Governing Equations
- Chapter 3: Finite Element Method
- Chapter 4: Computational Code
- Chapter 5: Validation
- Chapter 6: Results
- Conclusion

1 LITERATURE REVIEW

1.1 Introduction

In this chapter, the literature review is presented and discussed, in order to the problem and the methodology used in this work, such as the drug-eluting stents and the Finite Element Method applied to the convection-diffusion equation.

1.2 Drug-Eluting Stent

According to the World Health Organization (WHO) [?], cardiovascular diseases have remained the leading deaths causes globally in the last 15 years. It is estimated that 15.2 million people died from CVD in 2016, representing 26.7% of all deaths in the world. In Brazil, about 38% of deaths due to CVD is in the procutive age range (18 to 65 years) and the estimated costs of CVD were R$ 37.1 billion in 2015, that is, 0.7% GDP [?].

About 60% of CVD deaths occurred due to coronary artery disease (CAD). The main cause of CAD is the atherosclerosis which consists of the accumulation of fatty plaques inside the artery wall causing a decrease in lumen diameter. The Atherosclerosis can be prevented with a change in harmful habits such as: cigarette smoking, physical inactivity/low fitness and poor dietary habits [?]. For a corrective approach, however, two treatments can be performed: the *Coronary Artery Bypass Sugery* (CABG) or a procedure minimally invasive where a wire tube is placed. The procedure development originates in 1964 with Dotter and Judkins [?], where they introduced a new technique for obstructed femoral artery treatment due to atherosclerosis. This technique is known as *percutaneous transluminal angioplasty*. Such procedure is presented applicable to other arteries, including the coronary artery.

In 1979, Gruntzig, Senning and Seigenthaler [?] performed the percutaneous transluminal technique in the artery coronary artery using a balloon catheter in order to dilate the site with stenosis. The procedure was performed on 50 patients for 18 months and satisfactory results were presented, mainly with patients with only a single artery with sterosis. Such procedure is known as *Coronary Angioplasty Percutaneous Transluminal* (PTCA).

Although PTCA using a balloon has shown satisfactory results, over time, the artery presented restenosis. In 1987, Sigwart et al. [?] present the result of implanting a prosthesis made of a self-expanding stainless steel mesh in the femoral and coronary arteries of 25

patients who cases of restenosis. The prosthesis proved to be a interesting way to solve restenosis. This prosthesis was called *stent*.

In 1994, Serruys et al. [?] present a comparison between PTCA procedures using an balloon and stent implantation. 520 patients were analyzed, where 262 patients with implanted stents and 258 patients with the inflatable balloon. The clinical and angiographic results were better for the patients who had stent implantation to those who underwent the procedure only using the balloon. Thus, the PTCA procedure with stent implantation was confirmed as a more effective solution than PTCA using only the balloon. However, a problem persisted: the restenosis. During the 1990s, researchers sought to solve this problem. The Figure 1 presents the comparison between PTCA procedures using the balloon and stent.

(a)   (b)
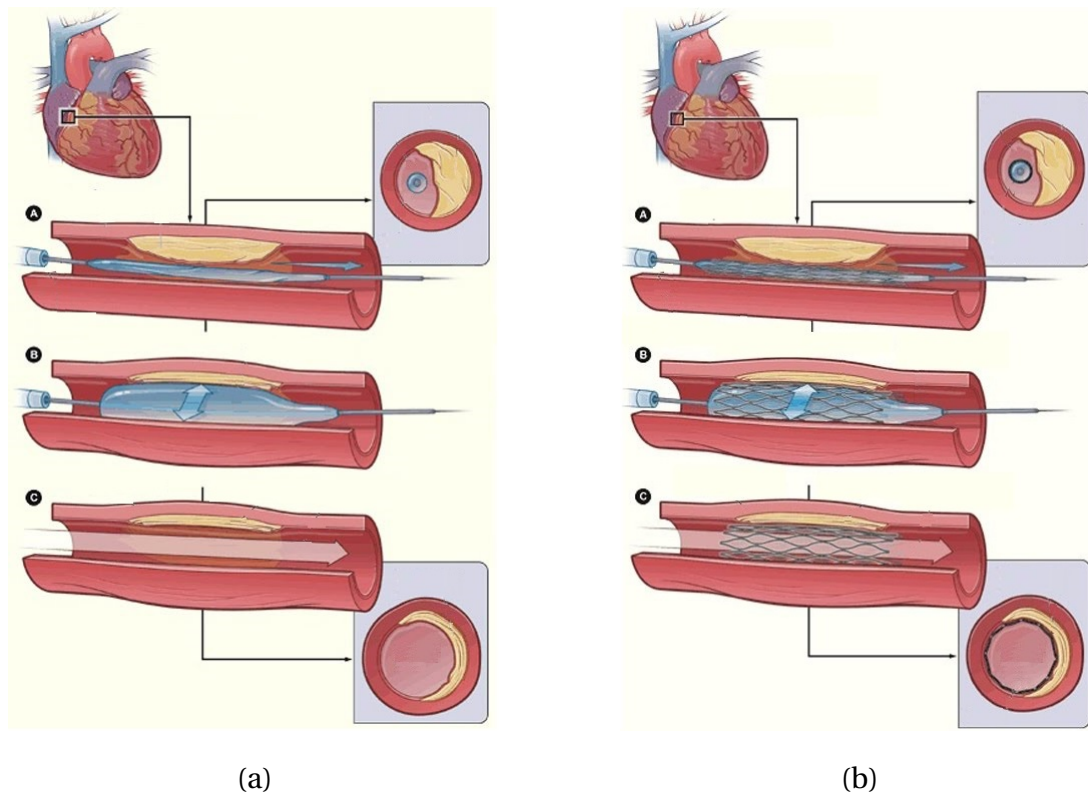
Figure 1: Comparison PTCA procedure: (a) balloon and (b) stent.

In 2001, Hwang, Wu and Edelman [?] presented a simulation of stent implantation coated with a drug in a coronary artery. The simulation presented the close relationship between drug distribution and *Peclet number* in addition to the importance of developing geometries for stents that enhance the diffusion of the chemical substance. Such procedure proved to be a promising option for the treatment of atherosclerosis and reestonosis. This new type of stent would be known as *drug-eluting stent*.

In 2009, Zunino et al. [?] presented a complete overview of mathematical models and finite element numerical simulation applied to the modelling of drug eluting stens and of their interaction with the coronary arteries, take into account the stent expansion, fluid dynamics around the stent and drug release. The numerical simulation shown recirculation zones downstream has important consequences on the drug release process. The smooth and concave shape of stent contours shows that part of the drug released and accumulated in the neighborbood of the links is transported away and may affect the arterial walls located downstream. However, for the case analyzed, the authors concluded that the drug released into the lumem does not significantly contribute to the permanent drug deposition into the arterial wall and only an small fraction of the total amount drug stored into the stent was effectively delivered to the artery.

In 2014, Bozsak, Chomaz and Barakat [?] propose a computational model of transport of the drugs *paclitaxel* and *sirolimus* on the artery wall. Such drugs are frequently used in drug-eluting stents. The model takes into account the structure in multilayer of the artery wall and these layers were modeled as porous media. Thus, the law of *Darcy* was used to simulate the flow within the layers of the artery. The simulation showed that the choice of the type of drug used is a crucial parameter in the creation of the drug-eluting stent due to transport in the artery wall.

In 2016, Bukac et al. [?] present a fluid-structure interaction between a curved coronary artery with an implanted stent, pulsatile blood flow and heart contractions. An finite element numerical simulation was performed using ALE approach and the Navier-Stokes equations for an incompressible, viscous fluid are used to model the blood flow. The performance of the four commercially available stent geometries stent struts was evaluated based on the pathobiologic parameters responses leading to restenosis in the curved coronary arteries and the horizontal sinusoidal of *Cypher stent struts* performed the best in terms of theses parameters. However, on limitation of the model used in the work is that it does not account for the protrusion of stent struts into the vessel lumen. Thus, the influence of small-scale vortices around stent struts on wall shear stress was not studied.

Recently, Wang et al. [?] present the simulation of blood flow in a coronary artery with atherosclerosis and drug-eluting stent placed. Blood is approximated as a Newtonian and monophase fluid and the governing equations were approximated according to the Finite Element Method. Several axisymmetric geometries were presented, including a real coronary

artery. Such geometries were used for this current work, but modified for a two-dimensional approach. The simulations showed that the proposed simplified artery with atherosclerosis model produced similar results of velocity, pressure and concentration when compared to the real artery.

The following year, Lucena et al. [?] present the simulation of the transport of the drug *sirolimus* on the wall of an artery modeled as a porous and anisotropic medium. Dissolution in the polymeric stent lining in addition to transport in the artery wall in an axysymmetric domain was considered. The government equations were approximated according to the Finite Element Method. The work showed that the evolution time of the transport process can be efficiently controlled by the mass diffusivity of the polymer. It is estimated that about 47% of the drug is diffused in the lumen and is lost in the bloodstream. The spatial distribution of the drug, however, is greatly influenced by blood flow and the properties of the artery wall. Thus, such results are susceptible to the patient's health conditions.

In 2019, Gudino, Oishi and Sequeira [?] presented the influence of non-Newtonian blood flow models on drug diffusion from a coronary drug-eluting stent. The Oldroyd-B, Phan-Thien-Tanner and Giesekus viscoelastic models were used to describe the fluid dynamics of blood and the finite element method was used for numerical simulations. The simulations shown the hemodynamics captured by each model are more significant in the proximal recirculation zones. The comparison between the newtonian and non-newtonian model were performed and the results of total stress tensor as well as the drug concentration in the artery wall showed significant differences between the models.

Over the past and current decade, several drug-eluting stents have been developed such as: *Ravel* [?], *Taxus I and Taxus II* [?] [?], *C-Sirius* [?], *Smart* [?] and more recent ones as presented in Figure 2. Currently, a new generation of stents has been developed in which the entire structure is absorbed. Such a generation is known as *bioabsorbable stent*, the use of this technology is not the subject of this work.
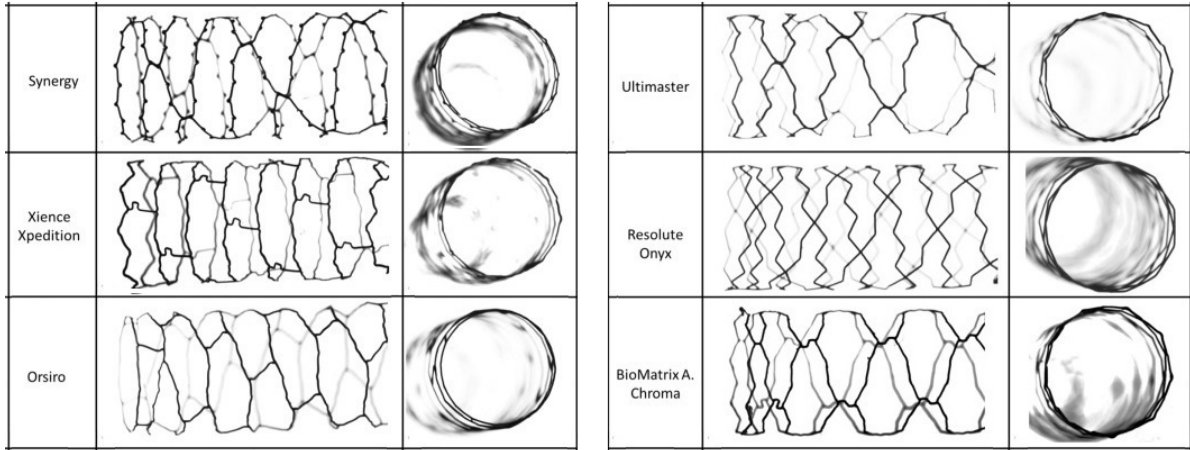
Figure 2: Several models of drug-eluting stent [**?**].

1.3 Finite Element Method - Convection-Diffusion Equation

The mathematical basis for the Finite Element Method begins in 1909 with Ritz [**?**] in which a continuous problem is replaced by a discrete problem with a finite number of degrees of freedom where the unknowns were approximated by the product between the constants and the base functions chosen in order to guarantee the accuracy of the result. This procedure is known as *variational formulation.*

Years later, Galerkin (1915) [**?**] uses the Weighted Residual Method to determine the constants of the variational formulation where the same base functions were used in the weight functions. This procedure is known as *Galerkin formulation* and is widely used nowadays.

During the 1940s, Courant (1943) [**?**] applied variational formulation to a domain discretized by triangular elements. In 1965, Zienkicwicz and Cheung [**?**] show that the Weighted Residual Method has a good approximation of the solution and the Finite Element Method was formalized to solve several problems. The proposed mathematical approach is often used today.

The Finite Element Method has become a very effective tool in the solution of several problems and it has been widely used in problems of the solids mechanics. In fluid mechanics, however, its use became possible only later due to the spurious oscillations that it can be seen when the convective term is superior to the diffusive term. Such oscillations are present not only in the Finite Element Method but were also observed in the Finite Difference Method

by Spalding in 1972 [?] where it is shown that the *upwind* effect helped to reduce these oscillations.

In 1976, Christie et al. [?] modify weight functions for asymmetric or quadratic functions to reduce spurious oscillations in one-dimensional diffusion-convection problems. Such modifications produced a *upwind* effect on the solution. This procedure became known as *Petrov Galerkin Formulation*. In the following year, Heinrich, Huyakorn and Zienkiewicz [?] generalize the scheme to a two-dimensional problem. The global matrices, however, became asymmetrical differently from those presented in the Galerkin scheme.

In 1982, Brooks and Hughes [?] proposed a new formulation that consists of the weight functions modified in order to the diffusion operator acts only in the flow direction. This procedure aims to eliminate the excess of diffusion perpendicular to the flow that the Petrov-Galerkin scheme presented in some cases. The formulation does not require the use of high-order weight functions and was efficient in eliminating perpendicular diffusion. The formulation received the name *Streamline Upwind Petrov-Galerkin* (SUPG).

In the same year, Pironneau [?] presented the *Characteristic trajectory* applied to the Finite Element Method in solving the non-steady convection-diffusion and Navier-Stokes equations. Thereby, the author was able to derive conservative schemes of the type *upwind* with first and second order accurate. As the matrices are symmetric, this scheme proved to be advantageous in solving linear systems compared to other *upwind* schemes. In addition to the method is unconditionally stable. The numerical implementation, however, requires an complex searching procedure. This method is known as *semi-Lagrangian* Method.

In 1984, Donea [?] presents an alternative for solving multidimensional and transient convection-diffusion problems. This alternative is known as the *Taylor-Galerkin* scheme. The scheme consists of using the high-order terms of the Taylor expansion to reduce spurious oscillations. Unlike upwind schemes, in the Taylor-Galerkin scheme there is no need to use modified weight functions. The scheme is compared with the formulations of Galerkin and Petrov Galerkin and it showed high accuracy and low numerical diffusion. Although the computational implementation is easy, the Taylor-Galerkin method is conditionally stable.

In the same year, Lohner et al. [?] proposed a simpler alternative to avoid the complex searching procedure in the characteristic methods. This alternative involves to use the high-order terms of the Taylor expansion to approximate the departure point. The procedure

is known as *Characteristic Galerkin.* The main advantages of method are the symmetric global matrices as well as no searching procedure for each time step. However, the method is conditionally stable. Although the Taylor Galerkin and Characteristic Galerkin discretization procedures are distinct, the system of equations is the same for the convection-diffusion equation, where the unknown is a scalar.

Several researchers have analyzed the stability and convergencei of these methods as the paper of Donea and Quartapelle (1992) [**?**]. In this paper, the authors present an analysis of several methods for solving unsteady problems governed by advection equations. Whereas the Petrov-Galerkin and SUPG schemes have the asymmetric matrix system, the Taylor-Galerkin, Characteristic Galerkin and semi-Lagrangian methods have the advantage of matrices were symmetric facilitating computational implementation. In addition to the Taylor-Galerkin and Characteristic Galerkin methods have a simpler implementation and they can achieve a high order accuracy, but they are only conditionally stable. For large time step, unlike, the semi-Lagrangian method is more efficient because its unconditional stability. However, the main disadvantage of this method is the complex searching procedure. It may lead to excessive computational cost if it is not well designed.

All the methods presented have satisfactory results and they are well known in the literature. These methods, therefore, made it possible to solve convective problems using the Finite Element Method and they are presents in several complex problems today, such as the numerical simulation for two-phase flows with dynamic boundaries presented by Anjos, Mangiavacchi and Thome (2020) [**?**]. For current work, the semi-Lagrangian method was chosen for decrease spurious oscillations, due to its unconditional stability and the symmetric linear system of the governing equation in an finite element context.

2 GOVERNING EQUATIONS

2.1 Introduction

In this work, the fluid is considered as a continuum body. This means that given an element of infinitesimal fluid, it is large enough that there are no empty spaces in its domain. Thereby, the flow can be modeled according to universal conservation principles such as:

- Mass Conservation

- Linear Momentum Conservation

- Species Transport Conservation

These are the principles that govern the flow proposed in this work. Initally, the Arbitrary Lagrangian-Eulerian description is presented. Then, in the section 2.3, we will present the principle of mass conservation and the *continuity equation* for an incompressible fluid. In the section 2.4, the *Navier Stokes equation* for an incompressible fluid is presented according to the principle of momentum linear conservation. In the section 2.5, we will present the *Species Transport Conservation*. Then, the governing equations are non-dimensionalization in the 2.6 section and the Navier-Stokes equation is presented according to *vorticity-streamfunction formulation* in the 2.7 section.

2.2 Arbitrary Lagrangian-Eulerian

The choice of the kinematical description of the continuum is extremely important for the development of a computational code, since it directly affects the accuracy of the numerical result. In the literature, two classic descriptions are commonly used, namely Lagrangian and Eulerian.

The Lagrangian description is one where each computational mesh node moves at the same velocity as the material point, as can be seen in Figure 3a. Thus, for each time step, we have a new computational mesh. The main advantage of this description is that the value of the computational mesh node will have the same value as the material point and thus, numerical diffusion will not be observed. In addition, the Lagrangian description makes it possible to perform fluid-structure interaction simulations. However, for large deformations,

it is necessary to implement an insetion and deletion node algorithm for computational mesh.

The Eulerian description is the one where the computational mesh node remains fixed for each time step, as can be seen in Figure 3b. Thus, the computational mesh node value will be an interpolation of the material point, causing the presence of numerical diffusion in the solution. However, the computational cost is relatively attractive since it is not necessary the remeshing in each time step.

A description, however, that combines the advantages of these two classic descriptions as well as minimizing their disadvantages would be most appropriate. In this context that the Arbitrary Lagrangian-Eulerian description (ALE) was developed. This description considers that the velocity field of computational mesh is unlike than the material point and null value, as can be seen in Figure 3c. In this way, it can be calculated as a linear combination of other velocity fields, so that we have an optimal relationship between numerical diffusion and mesh deformation. In addition, it is possible to assign several velocity values to specific regions of the problem in order to improve the accuracy of the solution.
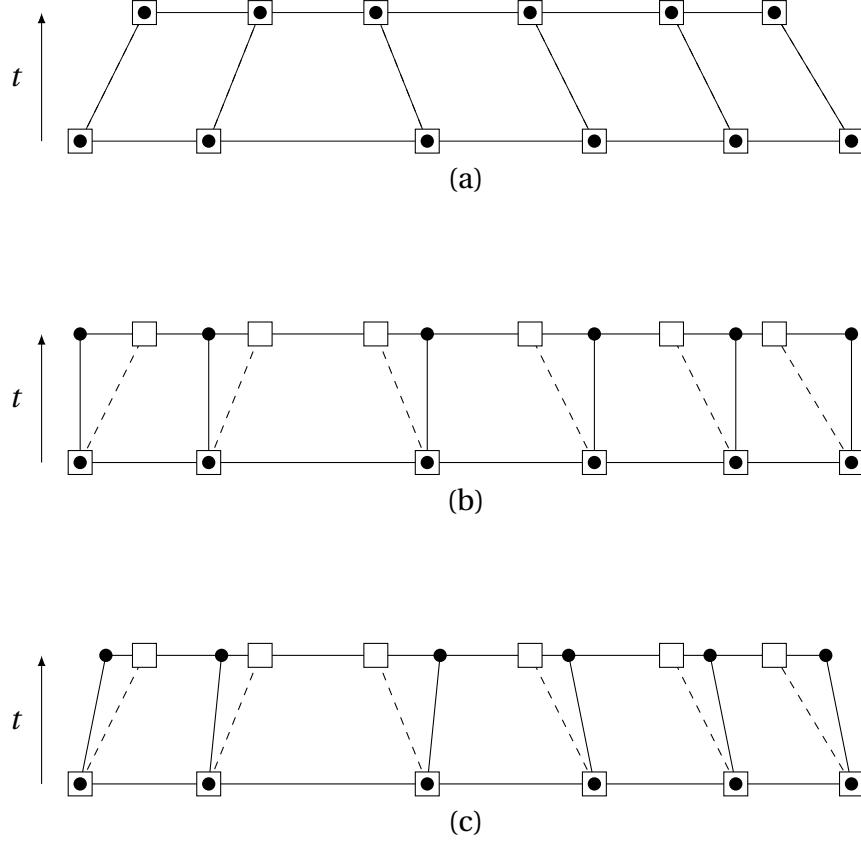
Figure 3: One-dimensional examples of the (a) Lagrangian description, (b) Eulerian description and (c) ALE description, where the white square is the material point, the black point is the mesh node, the dashed line is the characteristic trajectory and the continuous line is mesh nodes trajectory.

The ALE description was first implemented in the finite difference method, as presented by Hirt et al. (1974) [?] and was subsequently adopted in the finite elements context, as presented by Donea (1982) [?]. In this description, the referential domain that describes the computational mesh moving is different from the material domain and the spatial domain, as shown in the Figure 4. However, it is possible to correlate these frameworks. For instance, if the Φ operator is equal to the identity matrix ($\mathbf{I}$), then the referential and material domain is the same and, subsequently, the node velocity of the computational mesh is equivalent to the material points velocity (Lagrangian description). But, if the Ψ operator is equal to $\mathbf{I}$, the computational mesh velocity is equivalent to null value and then the Eulerian description is obtained. For more details, it is possible to consult the works of Donea (2004) [?] and Hughes (1981) [?].
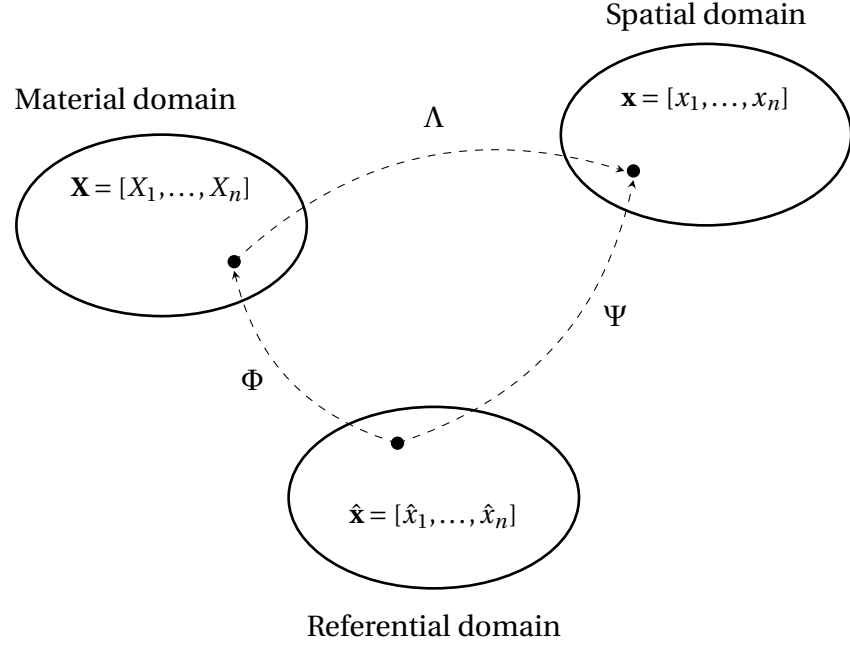
Figure 4: Material, Spatial and Referential domains for the Artibratiry Lagrangian-Eulerian description.

2.3 **Mass Conservation**

As presented by Panton (2013) [**?**], the mass conservation for a region with arbitrary motion is:

$$\int_V \frac{\partial \rho}{\partial t} dV + \oint_S \rho\,(\mathbf{v} - \hat{\mathbf{v}}) \cdot \mathbf{n} dA = 0 \tag{2.1}$$

where ρ is density, $\mathbf{v}$ is material velocity, $\hat{\mathbf{v}}$ the computational mesh velocity and the subtraction of these velocities is known as *relative velocity* $\mathbf{c}$. Applying the *Gauss theorem* on surface integral:

$$\int_V \frac{\partial \rho}{\partial t} dV = -\int_V \nabla \cdot \left(\rho \mathbf{c} \right) dV \tag{2.2}$$

that is:

$$\int_V \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{c}) \right] dV = 0 \tag{2.3}$$

In view of the $dV \neq 0$, the Eq. 2.3 can be presented as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{c}) = 0 \tag{2.4}$$

Developing the equation, we have:

$$\frac{\partial \rho}{\partial t} + \mathbf{c} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{v} = 0 \tag{2.5}$$

According to fluid incompressible assumption, the density does not depende on time and on coordinates. Therefore, the $\partial \rho / \partial t$ e $\nabla \rho$ derivatives are null values. Thus, the mass conservation is reduced to:

$$\rho \nabla \cdot \mathbf{v} = 0 \tag{2.6}$$

that is:

$$\nabla \cdot \mathbf{v} = 0 \tag{2.7}$$

This equation is known as the *continuity equation* for an incompressible flow in an Arbitrary Lagrangian-Eulerian description [**?**].

2.4 Linear Momentum Conservation

The same concept of mass conservation is applied to the linear momentum conservation. Therefore, the linear momentum conservation for a region with arbitrary motion

is:

$$\int_V \frac{\partial}{\partial t}(\rho \mathbf{v}) dV = -\oint_S \rho \mathbf{v} \mathbf{c} \cdot \mathbf{n} dA + \oint_S \sigma \cdot \mathbf{n} dA + \int_V \rho \mathbf{g} dV \tag{2.8}$$

where σ is the stress tensor and $\mathbf{g}$ is gravity vector. Applying the *Gauss Theorem* on surface integrals:

$$\int_V \frac{\partial}{\partial t}(\rho \mathbf{v}) dV = -\int_V \nabla \cdot (\rho \mathbf{v} \mathbf{c}) dV + \int_V \nabla \cdot \sigma dV + \int_V \rho \mathbf{g} dV \tag{2.9}$$

that is:

$$\int_V \left[\frac{\partial}{\partial t}(\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \mathbf{c}) - \nabla \cdot \sigma - \rho \mathbf{g} \right] dV = 0 \tag{2.10}$$

In view of the $dV \neq 0$, the Eq. 2.10 can be presented as:

$$\frac{\partial}{\partial t}(\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \mathbf{c}) - \nabla \cdot \sigma - \rho \mathbf{g} = 0 \tag{2.11}$$

that is:

$$\frac{\partial}{\partial t}(\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \mathbf{c}) = \nabla \cdot \sigma + \rho \mathbf{g} \tag{2.12}$$

Developing the left hand side of equation, we have:

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \frac{\partial \rho}{\partial t} + \rho \mathbf{c} \cdot \nabla \mathbf{v} + \mathbf{v} \nabla \cdot (\rho \mathbf{c}) = \rho \left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right] + \mathbf{v} \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{c}) \right] \tag{2.13}$$

The last term of above equation is null because the *continuity equation* (Eq. 2.4). Thus,

the linear momentum equation can be rewritten as:

$$\rho\left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c}\cdot\nabla\mathbf{v}\right] = \nabla\cdot\sigma + \rho\mathbf{g} \tag{2.14}$$

The stress tensor σ can be split into two tensors:

$$\sigma = -p\mathbf{I} + \tau \tag{2.15}$$

where, p is pressure field, $\mathbf{I}$ is the identity matrix and τ id deviatoric stress. Replacing them in Eq. 2.14, we have:

$$\rho\left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c}\cdot\nabla\mathbf{v}\right] = \nabla\cdot\left[-p\mathbf{I} + \tau\right] + \rho\mathbf{g} \tag{2.16}$$

that is:

$$\rho\left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c}\cdot\nabla\mathbf{v}\right] = -\nabla p + \nabla\cdot\tau + \rho\mathbf{g} \tag{2.17}$$

The deviatoric stress τ depends on strain tensor rate and we can define it relating to medium physical properties. Whereas a homogeneous, isotropic fluid and the deviatoric stress as a continuous and linear function of velocity gradient, we have:

$$\tau = \mu\left[\nabla\mathbf{v} + (\nabla\mathbf{v})^T\right] + \lambda\mathbf{I}\nabla\cdot\mathbf{v} \tag{2.18}$$

where μ is dynamic viscosity of fluid, λ is known as the second viscosity coefficient and $\mathbf{I}$ is identidy matrix. Replacing them in Eq. 2.17, we have:

$$\rho\left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c}\cdot\nabla\mathbf{v}\right] = -\nabla p + \nabla\cdot\left[\mu\left[\nabla\mathbf{v} + (\nabla\mathbf{v})^T\right] + \lambda\mathbf{I}\nabla\cdot\mathbf{v}\right] + \rho\mathbf{g} \tag{2.19}$$

that is:

$$\rho\left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{v}\right] = -\nabla p + \nabla \cdot \left[\mu\left[\nabla \mathbf{v} + (\nabla \mathbf{v})^T\right]\right] + \nabla \cdot \left[\lambda \mathbf{I} \nabla \cdot \mathbf{v}\right] + \rho \mathbf{g} \tag{2.20}$$

Taking into consideration that the dynamic viscosity μ does not depends on coordinates, we have:

$$\rho\left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{v}\right] = -\nabla p + \mu\left[\nabla \cdot \nabla \mathbf{v} + \nabla \cdot (\nabla \mathbf{v})^T\right] + \nabla \cdot \left[\lambda \mathbf{I} \nabla \cdot \mathbf{v}\right] + \rho \mathbf{g} \tag{2.21}$$

that is:

$$\rho\left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{v}\right] = -\nabla p + \mu\left[\nabla^2 \mathbf{v} + \nabla(\nabla \cdot \mathbf{v})\right] + \nabla \cdot \left[\lambda \mathbf{I} \nabla \cdot \mathbf{v}\right] + \rho \mathbf{g} \tag{2.22}$$

According to Eq. 2.7, we have:

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{v} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{g} \tag{2.23}$$

where ν is the kinematic viscosity of fluid. The Eq. 2.23 is known as *Navier-Stokes Equation* and is valid for a homogeneous, isotropic, incompressible fluid and with viscosity that it does not depends on coordinates, according to the Arbitrary Lagrangian-Eulerian description [?].

2.5 Species Transport Conservation

Similarly to the previous sections, the species transport conservation for a region with arbitrary motion is:

$$\int_V \frac{\partial e}{\partial t} dV = -\oint_S e\mathbf{c} \cdot \mathbf{n} dA + \oint_S D\nabla e \cdot \mathbf{n} dA + \int_V \dot{R} dV \tag{2.24}$$

where D is the mass diffusivity and $\dot{R}$ is the chemical species source rate. Applying the *Gauss Theorem* on surface integrals:

$$\int_V \frac{\partial e}{\partial t} dV = -\int_V \nabla \cdot (e\mathbf{c}) dV + \int_V \nabla \cdot (D\nabla e) dV + \int_V \dot{R} dV \tag{2.25}$$

that is:

$$\int_V \left[\frac{\partial e}{\partial t} + \nabla \cdot (e\mathbf{c}) - \nabla \cdot (D\nabla e) - \dot{R} \right] dV = 0 \tag{2.26}$$

In view of the $dV \neq 0$, the Eq. 2.26 can be represented by:

$$\frac{\partial e}{\partial t} + \nabla \cdot (e\mathbf{c}) - \nabla \cdot (D\nabla e) - \dot{R} = 0 \tag{2.27}$$

that is:

$$\frac{\partial e}{\partial t} + \nabla \cdot (e\mathbf{c}) = \nabla \cdot (D\nabla e) + \dot{R} \tag{2.28}$$

Developing the left hand side of equation, we have:

$$\frac{\partial e}{\partial t} + \mathbf{v} \cdot \nabla e + e\nabla \cdot \mathbf{c} = \nabla \cdot (D\nabla e) + \dot{R} \tag{2.29}$$

The last term of above equation is null because the fluid incompressibility assumption (Eq. 2.7), thus:

$$\frac{\partial e}{\partial t} + \mathbf{c} \cdot \nabla e = \nabla \cdot (D\nabla e) + \dot{R} \tag{2.30}$$

Taking into consideration that the mass diffusiviy is constant and without chemical

species generation, the species transport equation can rewritten as:

$$\frac{\partial e}{\partial t} + \mathbf{c} \cdot \nabla e = D\nabla^2 e \tag{2.31}$$

The Eq. 2.31 is known as *Species Transport Equation* for an incompressible fluid, with constant mass diffusivity and without chemical species generation for the Arbitrary Lagrangia-Eulerian description.

2.6 **Non-dimensionalization**

In this section, the non-dimensional form of continuity, Navier-Stokes and species transport equations are shown. The non-dimensionalization helps to understand which terms of the equation influence most during a given simulation in addition to allowing experiments with small scale models The following parameters was used in non-dimensionalization:

$$p = \rho_0 U^2 p^* \qquad e = (e_s - e_0)e^* + e_0 \qquad \nu = \nu_0 \nu^* \qquad D = D_0 D^* \qquad x = Lx^*$$

$$\mathbf{v} = U\mathbf{v}^* \qquad \mathbf{g} = g_0 \mathbf{g}^* \qquad \rho = \rho_0 \rho^* \qquad \nabla = \frac{1}{L}\nabla^* \qquad t = \frac{L}{U}t^*$$

where the asterisk identify the non-dimensional unknowns. Replacing the above parameters in Eq. 2.7, we have:

$$\frac{U}{L}\nabla^* \cdot \mathbf{v}^* = 0 \tag{2.32}$$

Multiplying both sides by U/L:

$$\nabla^* \cdot \mathbf{v}^* = 0 \tag{2.33}$$

A similar procedure is performed in Eq. 2.23, that is:

$$\frac{U^2}{L}\frac{\partial \mathbf{v}^*}{\partial t^*} + \frac{U^2}{L}\mathbf{c}^* \cdot \nabla^* \mathbf{v}^* = -\frac{U^2}{L}\frac{1}{\rho^*}\nabla^* p^* + \frac{\nu_0 U}{L^2}\nu^* \nabla^{*2}\mathbf{v}^* + g_0 \mathbf{g}^*$$ (2.34)

Multiplying both sides by L/U^2:

$$\frac{\partial \mathbf{v}^*}{\partial t^*} + \mathbf{c}^* \cdot \nabla^* \mathbf{v}^* = -\frac{1}{\rho^*}\nabla^* p^* + \frac{\nu_0}{UL}\nu^* \nabla^{*2}\mathbf{v}^* + \frac{g_0 L}{U^2}\mathbf{g}^*$$ (2.35)

that is:

$$\frac{\partial \mathbf{v}^*}{\partial t^*} + \mathbf{c}^* \cdot \nabla^* \mathbf{v}^* = -\nabla^* p^* + \frac{\nu_0}{UL}\nabla^{*2}\mathbf{v}^* + \frac{g_0 L}{U^2}\mathbf{g}^*$$ (2.36)

In the Eq. 2.31, a similar procedure is performed:

$$(e_s - e_0)\frac{U}{L}\frac{\partial e^*}{\partial t^*} + (e_s - e_0)\frac{U}{L}\mathbf{c}^* \cdot \nabla^* e^* = (e_s - e_0)\frac{D_0}{L^2}D^* \nabla^{*2} e^*$$ (2.37)

Multiplying both sides by $L/U(e_s - e_0)$, we have:

$$\frac{\partial e^*}{\partial t^*} + \mathbf{c}^* \cdot \nabla^* e^* = \frac{D_0}{UL}D^* \nabla^{*2} e^*$$ (2.38)

that is

$$\frac{\partial e^*}{\partial t^*} + \mathbf{c}^* \cdot \nabla^* e^* = \frac{D_0}{UL}\nabla^{*2} e^*$$ (2.39)

where, important non-dimensional groups are found in Eqs. 2.33, 2.36 and 2.39, such that: **Reynolds Number** (Re), **Froude number** (Fr) and **Mass Péclet number** (Pe_m). The Pe_m number is often shown as the product of the *Reynolds number* and the *Schmidt number Sc*.

Replacing these non-dimensional groups in Eqs. 2.33, 2.36 and 2.39 and removing the asterisk, we have:

$$\nabla \cdot \mathbf{v} = 0 \tag{2.40}$$

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{v} = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{v} + \frac{1}{Fr^2}\mathbf{g} \tag{2.41}$$

$$\frac{\partial e}{\partial t} + \mathbf{c} \cdot \nabla e = \frac{1}{ReSc}\nabla^2 e \tag{2.42}$$

The Eqs. 2.40, 2.41 e 2.42 are, respectively, non-dimensional form of Continuity, Navier-Stokes and Species Transport equations for a newtonian and incompressble flow in an Arbritary Lagrangian-Eulerian description.

2.7 **Vorticity-Streamfunction Formulation**

The Navier-Stokes equation has a strong coupling between the pressure field and the velocity field. This coupling makes it difficult to implement this equation computationally. The Decoupling of pressure and velocity fields is possible by using the *Vorticity-Streamfunction Formulation*. For this, we will replace in the equation 2.41 the following vector identity:

$$\mathbf{c} \cdot \nabla \mathbf{v} = \nabla \left(\mathbf{c} \cdot \mathbf{v} \right) - \mathbf{c} \times \nabla \times \mathbf{v} \tag{2.43}$$

Therefore:

$$\frac{\partial \mathbf{v}}{\partial t} + \nabla \left(\mathbf{c} \cdot \mathbf{v} \right) - \mathbf{c} \times \nabla \times \mathbf{v} = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{v} + \frac{1}{Fr^2}\mathbf{g} \tag{2.44}$$

Computing the curl on both sides of the above equation:

$$\nabla \times \frac{\partial \mathbf{v}}{\partial t} + \nabla \times \nabla (\mathbf{c} \cdot \mathbf{v}) - \nabla \times \mathbf{c} \times \nabla \times \mathbf{v} = -\nabla \times \nabla p + \frac{1}{Re} \nabla \times \nabla^2 \mathbf{v} + \frac{1}{Fr^2} \nabla \times \mathbf{g} \qquad (2.45)$$

that is:

$$\frac{\partial}{\partial t} \left[\nabla \times \mathbf{v} \right] + \nabla \times \nabla (\mathbf{c} \cdot \mathbf{v}) - \nabla \times \left[\mathbf{c} \times \nabla \times \mathbf{v} \right] = -\nabla \times \nabla p + \frac{1}{Re} \nabla^2 \left[\nabla \times \mathbf{v} \right] + \frac{1}{Fr^2} \nabla \times \mathbf{g} \qquad (2.46)$$

The terms that contain the gradient operator cancel each other out, since the curl of gradient of a scalar is zero. The last term is also null because the derivatives of a constant, as in the case of **g**, are equal to zero. Thus, we have:

$$\frac{\partial}{\partial t} \left[\nabla \times \mathbf{v} \right] - \nabla \times \left[\mathbf{c} \times \nabla \times \mathbf{v} \right] = \frac{1}{Re} \nabla^2 \left[\nabla \times \mathbf{v} \right] \qquad (2.47)$$

The vector $\nabla \times \mathbf{v}$ is known as *vorticity* (ω). Therby, the equation can be represented by:

$$\frac{\partial \omega}{\partial t} - \nabla \times \left[\mathbf{c} \times \omega \right] = \frac{1}{Re} \nabla^2 \omega \qquad (2.48)$$

The second term of left side in Eq. 2.48 can be replaced by following vectorial identity:

$$\nabla \times \left[\mathbf{c} \times \omega \right] = -\mathbf{c} \cdot \nabla \omega + \omega \cdot \nabla \mathbf{c} \qquad (2.49)$$

Thus the Eq. 2.48 will be:

$$\frac{\partial \omega}{\partial t} + \mathbf{c} \cdot \nabla \omega - \omega \cdot \nabla \mathbf{c} = \frac{1}{Re} \nabla^2 \omega \qquad (2.50)$$

For two-dimensional flows, as in the case of this work, the vorticity is perpendicular

to the velocity vector. Thus, the product $\omega \cdot \nabla \mathbf{c}$ will be canceled as presented by Pontes and Mangiavacchi (2016) [**?**]. Therefore:

$$\frac{\partial \omega}{\partial t} + \mathbf{c} \cdot \nabla \omega = \frac{1}{Re} \nabla^2 \omega \qquad (2.51)$$

The Eq. 2.51 is known as *vorticity equation* for two-dimensional flows of a Newtonian and incompressible fluid in an Arbritary Lagragian-Eulerian description. For a steady and two-dimensional flow of incompressible fluid, the velocity can be calculated from the volumetric flux. Thereby, the velocity is replaced by a scalar. Such a scalar is known as *streamfunction* (ψ). The relationship between the velocity components and the streamfunction is presented by expanding the continuity equation (Eq. 2.7):

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \qquad (2.52)$$

The following relationship between the streamfunction and the velocity components can be defined so that Eq. 2.52 is satisfied:

$$u = \frac{\partial \psi}{\partial y} \qquad v = -\frac{\partial \psi}{\partial x} \qquad (2.53)$$

In addition, the relationship between streamfunction and vorticity is shown expanding the $\nabla \times \mathbf{v}$ operation for the two-dimensional case:

$$\nabla \times \mathbf{v} = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \qquad (2.54)$$

Thus, replacing the Eq. 2.53 in Eq. 2.54, we have:

$$\omega = -\frac{\partial}{\partial x}\frac{\partial v}{\partial x} - \frac{\partial}{\partial y}\frac{\partial u}{\partial y} \qquad (2.55)$$

that is:

$$\omega = -\nabla^2 \psi \tag{2.56}$$

Therefore, the equations that govern the proposed problem in its non-dimensional form and vorticity-streamfunction formulation are shown below:

$$\frac{\partial w}{\partial t} + \mathbf{c} \cdot \nabla \omega = \frac{1}{Re} \nabla^2 \omega \tag{2.57}$$

$$\nabla^2 \psi = -\omega \tag{2.58}$$

$$\frac{\partial e}{\partial t} + \mathbf{c} \cdot \nabla e = \frac{1}{ReSc} \nabla^2 e \tag{2.59}$$

For further development using bt the semi-Lagragian Method, the equations above will be shown with the substantive derivative, thus:

$$\frac{D\omega}{Dt} = \frac{1}{Re} \nabla^2 \omega \tag{2.60}$$

$$\nabla^2 \psi = -\omega \tag{2.61}$$

$$\frac{De}{Dt} = \frac{1}{ReSc} \nabla^2 e \tag{2.62}$$

where material velocity field $\mathbf{v}$ is calculated by: $u = \partial \psi / \partial y$ and $v = -\partial \psi / \partial x$.

2.8 Generic Initial and Boundary Conditions

In numerical simulations, the choice of initial and boundary conditions is important to ensure result accuracy for any modeled problem by differential equations. The boundary conditions used are briefly explained below, followed by their detailed specifications in each particular case in the validations and results sections:

- *inflow condition*: this condition is specified when an mass inflow is desired. For such a

condition, $u = u_o$ and $v = v_o$.

- *wall condition*: this condition is specified at wall boundaries (moving wall and noslip conditions). All the velocity components are specified with the same wall velocity values.

- *outflow condition*: this condition represents a state where is close to a fully developed profile. Usually no value is specified for the unknowns.

- *free-slip condition*: this condition is specified at the symmetric axis. The normal velocity component is null and the derivative of the tangent component is also null value.

- *strut condition*: this condition is used on the stent. The normal and tangential velocity components are specified with null value. The concentration field is specified as $c = c_o$.

As mentioned by Batchelor (1967) [**?**], the ψ is constant along a streamline, then the streamfunction boundary condition can be calculated by $\psi_2 - \psi_1 = \int (u\,dy - v\,dx)$, where can be used u and v velocity inflow component. The ψ_1 and ψ_2 are usually called bottom and top streamlines, because the difference between two ψ values is equal to volume flow rate across inflow boundary. In this work, was set null value for bottom streamline. For top streamline, was calculated using u_o and v_o inflow velocity components, that is, $\psi_2 = \int (u_o\,dy - v_o\,dx)$.

The classical difficulty associated to the Streamfunction-Vorticity Formulation is due to the lack of vorticity boundary condition at a no-slip wall, as mentioned by Peyret (2013) [**?**]. Therefore, in this work, the vorticity boundary contition was calculated by $\omega = \nabla \times \mathbf{v}$ at each time step.

3 FINITE ELEMENT METHOD

3.1 **Introduction**

In this chapter, we will describe the Finite Element Method (FEM). The mathematical basis for the Finite Element Method begins in 1910s with Ritz [?] and Galerkin (1915) [?]. The proposal of the finite element procedure is an approximation applied to the terms of the variational formulation, for more details on the Finite Element Method see the works of Zienkiewicz and Taylor (2000) [?] and Hughes (2000) [?].

First, we will present the variational formulation with its the strong and weak form of the governing equations. Next, they are discretized in space using Galerkin Method with a linear triangular element and then the semi-Lagrangian Method is used to discretize them over time. Lastly, the matrix form of the governing equations are presented.

3.2 **Variational Formulation**

Governing equations in differential form with boundary conditions are known as **Strong Formulation**. Thus, the strong formulation for the proposed problem is:

$$\frac{D\omega}{Dt} = \frac{1}{Re}\nabla^2 w \tag{3.1}$$

$$\nabla^2 \psi = -\omega \tag{3.2}$$

$$\frac{De}{Dt} = \frac{1}{ReSc}\nabla^2 e \tag{3.3}$$

These equations are valid in $\Omega \subset \mathbb{R}^2$ domain with the following boundary conditions:

$$\begin{aligned}
\omega &= \omega_\Gamma \quad \text{in } \Gamma_1 \\
\psi &= \psi_\Gamma \quad \text{in } \Gamma_2 \\
e &= e_\Gamma \quad \text{in } \Gamma_4
\end{aligned} \tag{3.4}$$

The result of the weighted governing equations over the domain is known as **Weak formulation**. Then, the weak formulation for a single-phase, Newtonian and incompressible fluid using the vorticity-streamfunction formulation with species transport equation will be shown, for more details see the work of Brenner and Scott (1994) [**?**]. Therefore, as the objective is to find an approximate solution, it is acceptable to assume that a Residue **R** is produced in the governing equations, that is:

$$\frac{D\omega}{Dt} - \frac{1}{Re}\nabla^2\omega = R_1 \tag{3.5}$$

$$\nabla^2\psi + \omega = R_2 \tag{3.6}$$

$$\frac{De}{Dt} - \frac{1}{ReSc}\nabla^2 e = R_3 \tag{3.7}$$

We will force the residue to be equivalent to zero in a weighted sense, as mentioned by Finlayson (1972) [**?**], then:

$$\int_{\Omega} R_1 \cdot \delta \, d\Omega = 0 \tag{3.8}$$

$$\int_{\Omega} R_2 \cdot \phi \, d\Omega = 0 \tag{3.9}$$

$$\int_{\Omega} R_3 \cdot \eta \, d\Omega = 0 \tag{3.10}$$

where δ, ϕ and η are weight function. The weight functions are a set of arbitrary functions that belong to a function space that will be discussed later. We then have the following integrals:

$$\int_{\Omega} \left\{ \frac{D\omega}{Dt} - \frac{1}{Re}\nabla^2\omega \right\} \cdot \delta \, d\Omega = 0 \tag{3.11}$$

$$\int_{\Omega} \left\{ \nabla^2\psi + \omega \right\} \cdot \phi \, d\Omega = 0 \tag{3.12}$$

$$\int_{\Omega} \left\{ \frac{De}{Dt} - \frac{1}{ReSc} \nabla^2 e \right\} \cdot \eta \, d\Omega = 0 \tag{3.13}$$

Developing the integrals, we have:

$$\int_{\Omega} \frac{D\omega}{Dt} \delta \, d\Omega - \frac{1}{Re} \int_{\Omega} \nabla^2 \omega \delta \, d\Omega = 0 \tag{3.14}$$

$$\int_{\Omega} \nabla^2 \psi \phi \, d\Omega + \int_{\Omega} \omega \phi \, d\Omega = 0 \tag{3.15}$$

$$\int_{\Omega} \frac{De}{Dt} \eta \, d\Omega - \frac{1}{ReSc} \int_{\Omega} \nabla^2 e \eta \, d\Omega = 0 \tag{3.16}$$

In the diffusive term of the equations (3.14, 3.15 and 3.16), we will apply Green's theorem in order to decrease the derivative order and separate the term evaluated in the boundary. Thus the diffusive term will become:

$$-\frac{1}{Re} \int_{\Omega} \nabla^2 \omega \delta \, d\Omega = \frac{1}{Re} \int_{\Omega} \nabla \omega \cdot \nabla \delta \, d\Omega - \frac{1}{Re} \int_{\Gamma} \delta \nabla \omega \cdot \mathbf{n} \, d\Gamma \tag{3.17}$$

$$\int_{\Omega} \nabla^2 \psi \phi \, d\Omega = -\int_{\Omega} \nabla \psi \cdot \nabla \phi \, d\Omega + \int_{\Gamma} \phi \nabla \psi \cdot \mathbf{n} \, d\Gamma \tag{3.18}$$

$$-\frac{1}{ReSc} \int_{\Omega} \nabla^2 e \eta \, d\Omega = \frac{1}{ReSc} \int_{\Omega} \nabla e \cdot \nabla \eta \, d\Omega - \frac{1}{ReSc} \int_{\Gamma} \eta \nabla e \cdot \mathbf{n} \, d\Gamma \tag{3.19}$$

where $\mathbf{n}$ is the normal vector oriented outside the Γ. The last term in the above equations is known as a natural condition. As previously mentioned, the problem proposed in this work

has only Dirichlet conditions (known as an essential condition). Therefore, we will consider the assumptions $\delta = 0$, $\phi = 0$ and $\eta = 0$ in the equations (3.17, 3.18 and 3.19) for all Γ. Thus, the integral in Γ will be null and the diffusive term of the equations (3.14, 3.15 and 3.16) become:

$$-\frac{1}{Re}\int_\Omega \nabla^2 \omega \delta \, d\Omega = \frac{1}{Re}\int_\Omega \nabla\omega \cdot \nabla\delta \, d\Omega \tag{3.20}$$

$$\int_\Omega \nabla^2 \psi \phi \, d\Omega = -\int_\Omega \nabla\psi \cdot \nabla\phi \, d\Omega \tag{3.21}$$

$$-\frac{1}{ReSc}\int_\Omega \nabla^2 e\eta \, d\Omega = \frac{1}{ReSc}\int_\Omega \nabla e \cdot \nabla\eta \, d\Omega \tag{3.22}$$

Therefore, replacing the new diffusive terms in the governing equations:

$$\int_\Omega \frac{D\omega}{Dt}\delta \, d\Omega + \frac{1}{Re}\int_\Omega \nabla\omega \cdot \nabla\delta \, d\Omega = 0 \tag{3.23}$$

$$-\int_\Omega \nabla\psi \cdot \nabla\phi \, d\Omega + \int_\Omega \omega\phi \, d\Omega = 0 \tag{3.24}$$

$$\int_\Omega \frac{De}{Dt}\eta \, d\Omega + \frac{1}{ReSc}\int_\Omega \nabla e \cdot \nabla\eta \, d\Omega = 0 \tag{3.25}$$

If we consider that:

$$
\begin{aligned}
m_1(\frac{D\omega}{Dt},\delta) &= \int_\Omega \frac{D\omega}{Dt}\delta \, d\Omega & k_1(\omega,\delta) &= \int_\Omega \nabla\omega \cdot \nabla\delta \, d\Omega \\
m_2(\omega,\phi) &= \int_\Omega \omega\phi \, d\Omega & k_2(\psi,\phi) &= \int_\Omega \nabla\psi \cdot \nabla\phi \, d\Omega \\
m_3(\frac{De}{Dt},\eta) &= \int_\Omega \frac{De}{Dt}\eta \, d\Omega & k_3(e,\eta) &= \int_\Omega \nabla e \cdot \nabla\eta \, d\Omega
\end{aligned}
\tag{3.26}
$$

The equations, thus, can be shown in the weak form respectively:

$$m_1(\frac{D\omega}{Dt},\delta) + \frac{1}{Re}k_1(\omega,\delta) = 0 \tag{3.27}$$

$$-k_2(\psi,\phi) + m_2(\omega,\phi) = 0 \tag{3.28}$$

$$m_3(\frac{De}{De},\eta) + \frac{1}{ReSc}k_3(e,\eta) = 0 \tag{3.29}$$

Assuming that the sets of basis functions are:

$$
\begin{aligned}
&\mathbb{W} = \{\omega \in \Omega \to \mathbb{R}^2 : \int_\Omega \omega^2 d\Omega < \infty; \omega = \omega_\Gamma\} \\
&\mathbb{P} = \{\psi \in \Omega \to \mathbb{R}^2 : \int_\Omega \psi^2 d\Omega < \infty; \psi = \psi_\Gamma\} \\
&\mathbb{E} = \{e \in \Omega \to \mathbb{R}^2 : \int_\Omega e^2 d\Omega < \infty; e = e_\Gamma\}
\end{aligned}
\tag{3.30}
$$

and the set of weight functions space:

$$
\begin{aligned}
&\mathbb{D} = \{\delta \in \Omega \to \mathbb{R}^2 : \int_\Omega \delta^2 d\Omega < \infty; \delta_\Gamma = 0\} \\
&\mathbb{F} = \{\phi \in \Omega \to \mathbb{R}^2 : \int_\Omega \phi^2 d\Omega < \infty; \phi_\Gamma = 0\} \\
&\mathbb{N} = \{\eta \in \Omega \to \mathbb{R}^2 : \int_\Omega \eta^2 d\Omega < \infty; \eta_\Gamma = 0\}
\end{aligned}
\tag{3.31}
$$

The weak formulation consists to find the solutions of $\omega \in \mathbb{W}$, $\psi \in \mathbb{P}$, and $e \in \mathbb{E}$ such that:

$$m_1(\frac{D\omega}{Dt},\delta) + \frac{1}{Re}k_1(\omega,\delta) = 0 \tag{3.32}$$

$$-k_2(\psi,\phi) + m_2(\omega,\phi) = 0 \tag{3.33}$$

$$m_3(\frac{De}{De},\eta) + \frac{1}{ReSc}k_3(e,\eta) = 0 \tag{3.34}$$

for all $\delta \in \mathbb{D}$, $\phi \in \mathbb{F}$ and $\eta \in \mathbb{N}$.

3.3 **The semi-discrete Galerkin Method**

The weight functions δ, ϕ and η are sets of arbitrary functions and it have a large number of choices. In this work to discretize the domain, we will use the **Galerkin Method** which considers the same shape functions for the weight and interpolation functions of the unknowns (w, ψ, e). Thus, Eqs. 3.23, 3.24 and 3.25 can be presented in expanded form such as:

$$\int_\Omega \frac{D\omega}{Dt}\delta\, d\Omega + \frac{1}{Re}\int_\Omega \left\{\frac{\partial\omega}{\partial x}\frac{\partial\delta}{\partial x} + \frac{\partial\omega}{\partial y}\frac{\partial\delta}{\partial y}\right\}d\Omega = 0 \tag{3.35}$$

$$-\int_\Omega \left\{\frac{\partial\psi}{\partial x}\frac{\partial\phi}{\partial x} + \frac{\partial\psi}{\partial y}\frac{\partial\phi}{\partial y}\right\}d\Omega + \int_\Omega \omega\phi\, d\Omega = 0 \tag{3.36}$$

$$\int_\Omega \frac{De}{Dt}\eta\, d\Omega + \frac{1}{ReSc}\int_\Omega \left\{\frac{\partial e}{\partial x}\frac{\partial\eta}{\partial x} + \frac{\partial e}{\partial y}\frac{\partial\eta}{\partial y}\right\}d\Omega = 0 \tag{3.37}$$

We will now discretize the domain in *ne* elements and *np* nodes, where *ne* represents the total number of elements and *np* the total number of nodes in the computational mesh. Thus, we have:

$$\omega(\mathbf{x}, t) \simeq \sum_{i=1}^{np} \omega_i(t)N_i(\mathbf{x}) \tag{3.38}$$

$$\psi(\mathbf{x}, t) \simeq \sum_{i=1}^{np} \psi_i(t)N_i(\mathbf{x}) \tag{3.39}$$

$$e(\mathbf{x}, t) \simeq \sum_{i=1}^{np} e_i(t)N_i(\mathbf{x}) \tag{3.40}$$

where $\omega_i = [\omega_1, ..., \omega_{np}]$, $\psi_i = [\psi_1, ..., \psi_{np}]$ and $e_i = [c_1, ..., c_{np}]$ are the semi-discrete unknowns, that is, these unknowns are continuous in time (t) and discrete in space ($\mathbf{x}$). As they are only time dependent, then they can leave the integrals on the domain Ω. In addition, $N_i = [N_1, ..., N_{np}]$ are the approximation functions known as basis functions or interpolation functions. These functions can be chosen arbitrarily, however they must respect the boundary

conditions. They vary depending on the element type used for discretization. As mentioned, in this work we will use the same type of element for each governing equation, so we will have the same basis functions for all equations.

In Galerkin formulation, the weight functions assume the values of the basis functions, that is:

$$\delta(\mathbf{x}, t) \simeq \sum_{j=1}^{np} \delta_j(t) N_j(\mathbf{x}) \tag{3.41}$$

$$\phi(\mathbf{x}, t) \simeq \sum_{j=1}^{np} \phi_j(t) N_j(\mathbf{x}) \tag{3.42}$$

$$\eta(\mathbf{x}, t) \simeq \sum_{j=1}^{np} \eta_j(t) N_j(\mathbf{x}) \tag{3.43}$$

Thus, the governing equations in variational form discretized in space will be:

$$\int_\Omega \sum_{i=1}^{np} \frac{D\omega_i}{Dt} N_i \sum_{j=1}^{np} \delta_j N_j d\Omega$$
$$+ \frac{1}{Re} \int_\Omega \left\{ \sum_{i=1}^{np} \frac{\partial \omega_i N_i}{\partial x} \sum_{j=1}^{np} \frac{\partial \delta_j N_j}{\partial x} + \sum_{i=1}^{np} \frac{\partial \omega_i N_i}{\partial y} \sum_{j=1}^{np} \frac{\partial \delta_j N_j}{\partial y} \right\} = 0 \tag{3.44}$$

$$-\int_\Omega \left\{ \sum_{i=1}^{np} \frac{\partial \psi_i N_i}{\partial x} \sum_{j=1}^{np} \frac{\partial \phi_j N_j}{\partial x} + \sum_{i=1}^{np} \frac{\partial \psi_i N_i}{\partial y} \sum_{j=1}^{np} \frac{\partial \phi_j N_j}{\partial y} \right\} d\Omega$$
$$+ \int_\Omega \sum_{i=1}^{np} \omega_i N_i \sum_{j=1}^{np} \phi_j N_j d\Omega = 0 \tag{3.45}$$

$$\int_\Omega \sum_{i=1}^{np} \frac{De_i}{Dt} N_i \sum_{j=1}^{np} \eta_j N_j d\Omega$$
$$+ \frac{1}{ReSc} \int_\Omega \left\{ \sum_{i=1}^{np} \frac{\partial e_i N_i}{\partial x} \sum_{j=1}^{np} \frac{\partial \eta_j N_j}{\partial x} + \sum_{i=1}^{np} \frac{\partial e_i N_i}{\partial y} \sum_{j=1}^{np} \frac{\partial \eta_j N_j}{\partial y} \right\} d\Omega = 0 \tag{3.46}$$

Moving the sum symbols out of integrals, we have:

$$\sum_{j=1}^{np} \delta_j \left[\sum_{i=1}^{np} \frac{D\omega_i}{Dt} \int_\Omega N_i N_j d\Omega \right.$$
$$\left. + \sum_{i=1}^{np} \omega_i \left[\frac{1}{Re} \int_\Omega \left\{ \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right\} d\Omega \right] \right] = 0 \tag{3.47}$$

$$\sum_{j=1}^{np} \phi_j \left[-\sum_{i=1}^{np} \psi_i \int_\Omega \left\{ \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right\} d\Omega + \sum_{i=1}^{np} \omega_i \int_\Omega N_i N_j d\Omega \right] = 0 \tag{3.48}$$

$$\sum_{j=1}^{np} \eta_j \left[\sum_{i=1}^{np} \frac{De_i}{Dt} \int_\Omega N_i N_j d\Omega \right.$$
$$\left. + \sum_{i=1}^{np} e_i \left[\frac{1}{ReSc} \int_\Omega \left\{ \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right\} d\Omega \right] \right] = 0 \tag{3.49}$$

Taking into account that $\sum_{j=1}^{np} \delta_j \neq 0$, $\sum_{j=1}^{np} \phi_j \neq 0$ and $\sum_{j=1}^{np} \eta_j \neq 0$, then the governing equations discretized by Galerkin method are:

$$\sum_{j=1}^{np} \sum_{i=1}^{np} \left[\frac{D\omega_i}{Dt} \int_\Omega N_i N_j d\Omega + \omega_i \frac{1}{Re} \int_\Omega \left\{ \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right\} d\Omega \right] = 0 \tag{3.50}$$

$$\sum_{j=1}^{np} \sum_{i=1}^{np} \left[-\psi_i \int_\Omega \left\{ \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right\} d\Omega + \omega_i \int_\Omega N_i N_j d\Omega \right] = 0 \tag{3.51}$$

$$\sum_{j=1}^{np} \sum_{i=1}^{np} \left[\frac{De_i}{Dt} \int_\Omega N_i N_j d\Omega + e_i \frac{1}{ReSc} \int_\Omega \left\{ \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right\} d\Omega \right] = 0 \tag{3.52}$$

3.4 Mesh Elements

In the Finite Element context, the domain is discretized into several elements forming a computational mesh and the governing equations are applied to each of these elements.

The computational mesh can be structured or unstructured and its the choice is of vital importance for a good accuracy of the solution. Moreover, some parameters can influence the choice of a certain elements group, for instance in the case where there is a restriction condition as found in the Navier-Stokes equation due to the strong coupling between velocity and pressure fields. This restriction is known as *Babuska-Brezzi* [?] [?]. When we have this restriction, we need to have different numbers of nodes for each unkowns in the same element in order to have stability in the solution. Therefore, we need to use a *quadratic or cubic elements*. This methodology is known as *Mixed Finite Element Method*. But the vorticity-streamfunction formulation satisfies the *Babuska-Brezzi* restriction since there is no coupling between velocity and pressure fields. Thus, the use of a *linear element* does not produce instability and can be used without problems in this work.

Some triangular elements are presented with different orders of the interpolator polynomial below:

Linear Triangular Element: Due to its simplicity, it is the element most commonly used element in FEM when we have no restrictions. The analytical elementary matrices of this element are easily found in the literature. Since it is a linear element, the interpolation polynomial is first order. In this way, its interpolation functions are plane. This element is represented by the Figure 5:

$$N_i = L_i \qquad i = 1, 2, 3$$

Figure 5: Linear Triangular Element

Quad Triangular Element: This element is generally used when we have restrictions that prevent the use of the linear element or when we are interested in a better accuracy of the result. The elementary matrices of this element are calculated using the Gaussian Quadrature whose parameters can be found in the literature. Since it is a quadratic element, the interpolation polynomial is second order. In this way, its interpolation functions are parabolic. This
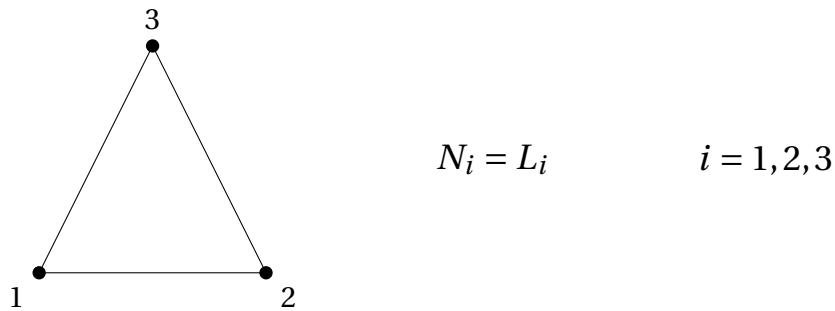
element is represented by Figure 6:

$$N_i = L_i(2L_i - 1)$$
$$N_4 = 4L_1 L_3$$
$$N_5 = 4L_2 L_3 \qquad i = 1, 2, 3$$
$$N_6 = 4L_3 L_1$$
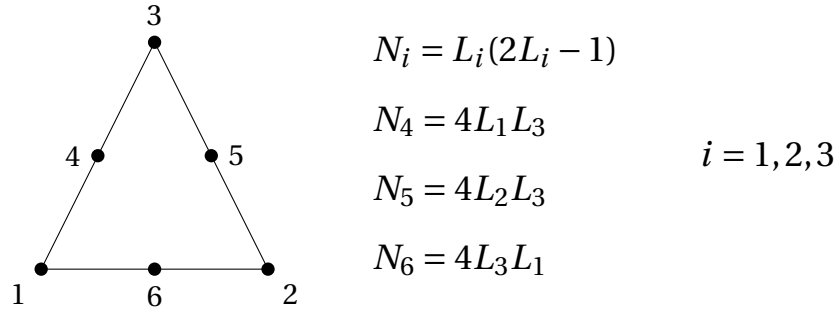
Figure 6: Quad Triangular Element

Mini Triangular Element: This element is used when we have restrictions that prevent the use of the linear element. Their elementary matrices are also calculated using the Gaussian Quadrature. Although it is an incomplete cubic element, the interpolation polynomial is still of third order. In this way, its interpolation functions have a bubble in the center of the element. This element is represented by Figure 7:

$$N_i = L_1 - 9L_1 L_2 L_3$$
$$N_4 = 27 L_1 L_2 L_3 \qquad i = 1, 2, 3$$
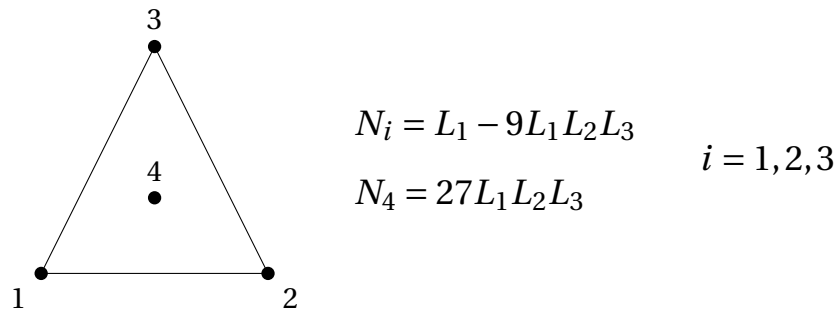
Figure 7: Mini Triangular Element

Triangular elements are the most common in 2D-FE Method because it allows a good discretization of irregular surfaces due to its geometric simplicity. In this work, we use a triangular element with the interpolator polynomial of order one, that is, linear. Therefore,

the elementary matrices can be defined by:

$$
\begin{aligned}
m^e &= \int_{\Omega^e} N_i^e N_j^e \, d\Omega \\
k_{xx}^e &= \int_{\Omega^e} \frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} \\
k_{yy}^e &= \int_{\Omega^e} \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y}
\end{aligned}
\tag{3.53}
$$

where the m^e, k_{xx}^e and k_{yy}^e are mass and stiffness elementary matrices respectively and the size is 3×3 for the linear triangular element. The contribution of each elementary matrix is guaranteed by the assembly operator **A** that it assembles the elementary matrix in global matrix, satisfying the local and global matrix index correspondence. Thus, the global matrix are defined as:

$$
M = \mathbf{A} m^e
\tag{3.54}
$$

$$
K_{xx} = \mathbf{A} k_{xx}^e
\tag{3.55}
$$

$$
K_{yy} = \mathbf{A} k_{yy}^e
\tag{3.56}
$$

where the M, K_{xx} and K_{yy} are mass and stiffness global matrices respectively and the size is $np \times np$, that is, node number by node number. Finally, the governing equations in matrix form according to the Finite Element Method that we used in this work were:

$$
M \frac{D\omega}{Dt} + \frac{1}{Re} \Big[K_{xx} + K_{yy} \Big] \omega = 0
\tag{3.57}
$$

$$
- \Big[K_{xx} + K_{yy} \Big] \psi + M\omega = 0
\tag{3.58}
$$

$$
M \frac{De}{Dt} + \frac{1}{ReSc} \Big[K_{xx} + K_{yy} \Big] e = 0
\tag{3.59}
$$

3.5 **The semi-Lagrangian Method**

For the numerical simulation of the non-steady convection-diffusion equation in an Finite Element context, several methods can be chosen, such as upwind, Taylor series in time and the characteristic trajectory as is the case of the semi-Lagrangian.

The semi-Lagrangian method was initially proposed by Sawyer (1963) [?] for a numerical simulation of atmospheric flow in a finite difference context. However, it was only in the 1980s that the semi-Lagrangian Method was presented according to the Finite Element approach by Pironneau (1982) [?], where it is shown the unconditional stability of the method and a symmetric linear systems to solve.

The material derivate of the Eqs. 3.57 and 3.59 may be discretized in the time domain at the x_i node by an explicit first order scheme:

$$\frac{D\omega}{Dt} \approx \frac{\omega_i^{n+1} - \omega_d^n}{\Delta t} \tag{3.60}$$

$$\frac{De}{Dt} \approx \frac{e_i^{n+1} - e_d^n}{\Delta t} \tag{3.61}$$

where, the variable Δt is time step, ω_i^{n+1} and e_i^{n+1} are the vorticity and concentration fields calculated in current time step at the current node position and ω_d^n and e_d^n are the vorticity and concentration fields calculated in previous time step at the departure node position. The departure node is calculated by solving characteristic equation:

$$\frac{d\mathbf{x}_d}{dt} = \mathbf{c} \tag{3.62}$$

where, $\mathbf{c}$ is the relative velocity and t is time variable and it is varies between $t \in \left[t^n, t^{n+1}\right]$. Assuming that in the current time the mesh computational nodes are known, so the initial condition for the Eq. 3.62 can be represented by $\mathbf{x}_d(t^{n+1}) = \mathbf{x}_i^{n+1}$, thus integrating both sides

of the equation:

$$\mathbf{x}_d^{n+1} - \mathbf{x}_d^n = \int_{t^n}^{t^{n+1}} \mathbf{c} dt, \qquad (3.63)$$

that is:

$$\mathbf{x}_d^n = \mathbf{x}_i^{n+1} - \int_{t^n}^{t^{n+1}} \mathbf{c} dt, \qquad (3.64)$$

Considering the relative velocity picewise constant in time, the departure point can be calculated by:

$$\mathbf{x}_d^n = \mathbf{x}_i^{n+1} - \mathbf{c}\Delta t, \qquad (3.65)$$

The Figure 8 shown the 1-dimensional characteristic trajectory of the material point in a moving computational mesh, where the white square is the material point and the black point is computational mesh. The dashed line is the characteristic trajectory that is represented by the Eq. 3.65. According to this scheme, the initial point x_i in t^{n+1} is known and therefore it is used to find the departure point x_d position in t^n.
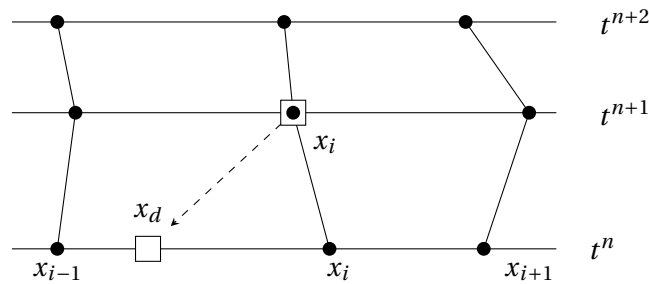

Figure 8: An one-dimensional space scheme where the departure node x_d is found by the characteristic trajectory in a moving mesh.

The main advantages of the semi-Lagrangian Method are the unconditional stability and the symmetric linear system to solve. Hovewer, the method has a disadvantage: the

searching procedure. The searching procedure in a 2-dimensional context may lead to excessive computational cost if it is not well designed. In this work, this iterative procedure is implemented using the 1-ring neighbors of the initial node, where the barycentric coordinates system is used in each neighbor element. If it is not found, the procedure continues with the nearest neighbor node until it finds the departure point. This procedure has presented an acceptable computational cost, in addition to the advantage of finding the nodes even in concave domains as can be seen in the Figure 9:
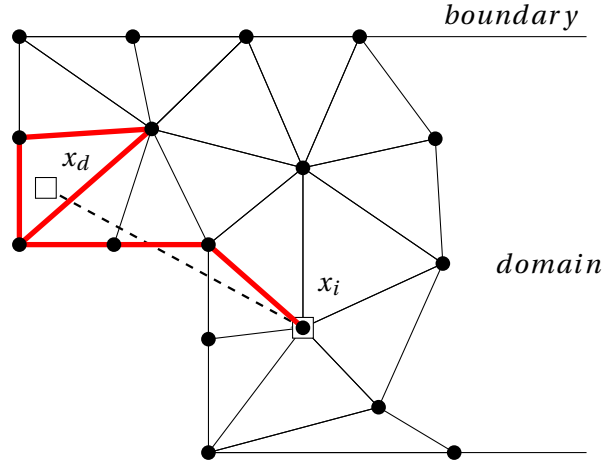

Figure 9: The searching procedure in a 2-dimensional concave domain, where the dashed line is the characteristic trajectory of 1st-order, the red line is the searching procedure path, the x_d is the departure point and the x_i is the initial point.

During the searching procedure, three situations may occur depending on the trajectory, as shown in Figure 10: the first and the second situations are similar, differentiating only the trajectory length. In the first situation, the departure node is inside near element from current node, while the second situation the departure node is inside far element from current node. The third situation, the departure node is outside domain then the vorticity and concentration fields receive the boundary condition value of nearest node to departure node.
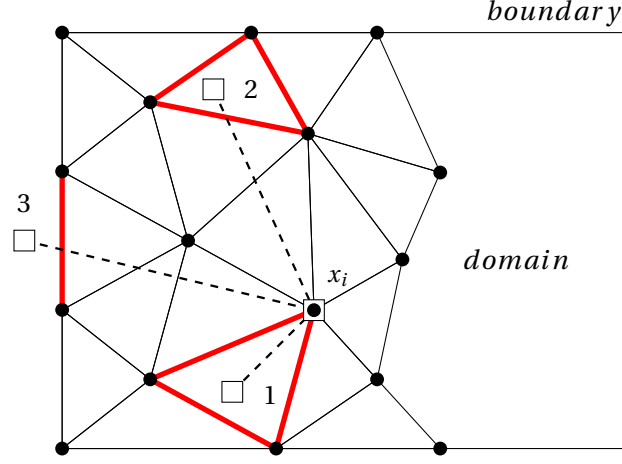
Figure 10: A two-dimensional space scheme where three situations may occur in the searching procedure.

After the departure point is found, the vorticity (ω_d) and concentration (e_d) fields are interpolated by the shape functions presented in the section 3.4. This interpolation procedure, which is typical in the Eulerian and Arbitrary Lagrangian-Eulerian descriptions, causes the numerical diffusion. To decrease the numerical diffusion, high-order elements can be used as the quadratic or cubic elements. Another way is to improve the discretization accuracy of the Eqs. 3.60 and 3.61. Therefore, the Eqs. 3.57, 3.58 and 3.59 can be shown in an implicit semi-Lagragian discretization as:

$$M\left[\frac{\omega_i^{n+1} - \omega_d^n}{\Delta t}\right] + \frac{1}{Re}\left[K_{xx} + K_{yy}\right]\omega_i^{n+1} = 0 \qquad (3.66)$$

$$-\left[K_{xx} + K_{yy}\right]\psi + M\omega = 0 \qquad (3.67)$$

$$M\left[\frac{e_i^{n+1} - e_d^n}{\Delta t}\right] + \frac{1}{ReSc}\left[K_{xx} + K_{yy}\right]e_i^{n+1} = 0 \qquad (3.68)$$

that is,

$$\frac{M}{\Delta t}\omega_i^{n+1} + \frac{1}{Re}\left[K_{xx} + K_{yy}\right]\omega_i^{n+1} = \frac{M}{\Delta t}\omega_d^n \tag{3.69}$$

$$\left[K_{xx} + K_{yy}\right]\psi = M\omega \tag{3.70}$$

$$\frac{M}{\Delta t}e_i^{n+1} + \frac{1}{ReSc}\left[K_{xx} + K_{yy}\right]e_i^{n+1} = \frac{M}{\Delta t}e_d^n \tag{3.71}$$

Therefore, the Vorticity-Streamfunction Formulation with Species Transport Equation discretized by *Galerkin* and *semi-Lagrangian Methods* in an ALE-FE context can be presented in matrix form as:

$$\left[\frac{M}{\Delta t} + \frac{1}{Re}\left[K_{xx} + K_{yy}\right]\right]\omega_i^{n+1} = \frac{M}{\Delta t}\omega_d^n \tag{3.72}$$

$$\left[K_{xx} + K_{yy}\right]\psi = M\omega \tag{3.73}$$

$$\left[\frac{M}{\Delta t} + \frac{1}{ReSc}\left[K_{xx} + K_{yy}\right]\right]e_i^{n+1} = \frac{M}{\Delta t}e_d^n \tag{3.74}$$

whereas material velocity field **v** is calculated by: $u = G_y\psi$ and $v = -G_x\psi$, where G_y and G_x are the *Gradient global matrix*.

4 COMPUTATIONAL CODE

4.1 Introduction

In this chapter, we will present the main characteristics of the computational code developed in Python 2.7 [?] using the object-oriented paradigm (OOP) in order to reuse the code in other simulations in the future. Initially, the UML diagram is shown by Figure 11 and the code framework is presented. In addition, the average processing time for each process is shown. Lastly, the Laplacian smoothing approach is presented with its applicability.

4.2 Code Framework

In order to reuse the numerical code in further simulations, the object-oriented paradigm (OOP) was used. In the Figure 11, the *UML Diagram* of the code framework is shown.
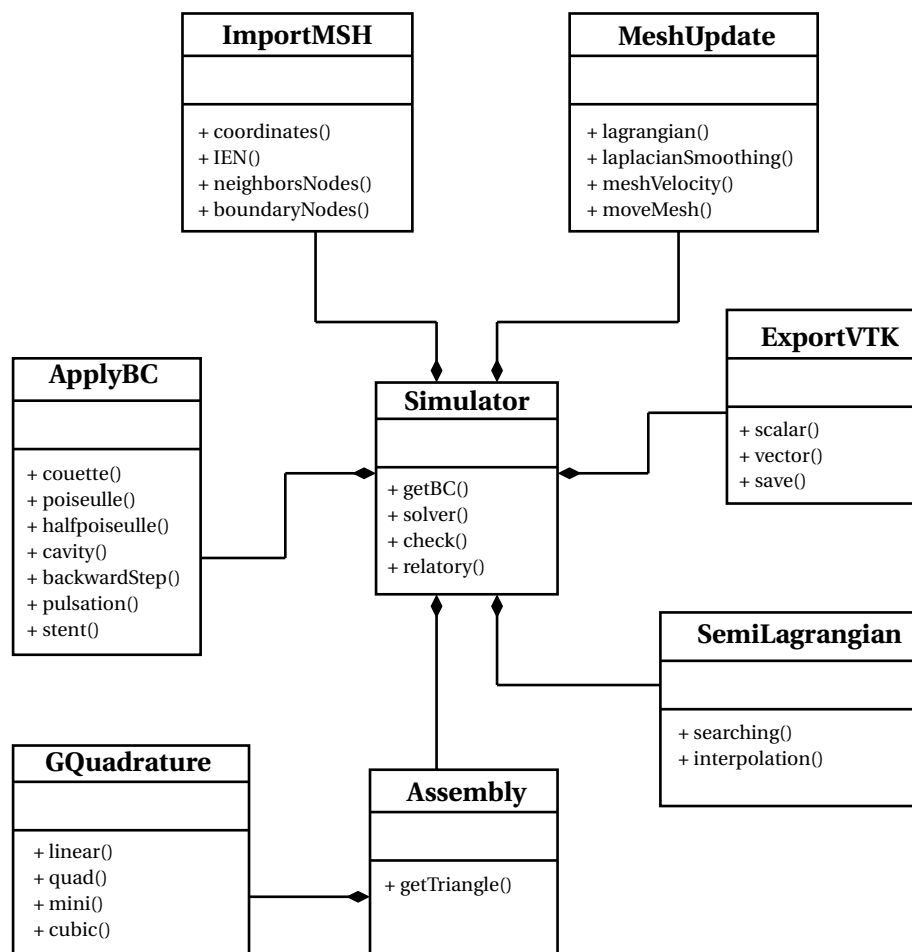

Figure 11: Simplified Class Diagram

Initially, the triangular unstructured mesh generated by GMSH [**?**] was imported into the numerical code by the *ImportMSH* class. This class returns important information for the simulation such as: *nodes number* (**np**), *elements number* (**ne**), *the coordinate vectors* (**x** and **y**), *the connectivity matrix* (**IEN**), *the neighbors nodes* and *the Boundary nodes*. The *ImportMSH* class is enabled to import the linear, quadratic or cubic triangular elements. The Table 1 shows the average processing time for mesh import in several unstructured linear triangular meshes.

Nodes	Elements	AVG Processing Time (s)
3436	6670	0.3
13745	26548	1.2
53229	105656	4.7
148187	295036	14.2

Table 1: Average processing time for mesh import in several unstructured linear triangular meshes

After importing the *.msh* file, the elementary and global matrices were assembled. The elementary matrices are created by the *GQuadrature* class, where it is enabled to assemble the linear, quadratic or cubic triangular element. For the linear triangular element, it is also possible to use elementary analytical matrices. For more details consult the work of Lewis, Nithiarasu and Seetharamu (2004) [**?**].

The global matrix assembly was performed by *Assembly* class, satisfying the local and global matrices index correspondence. They were initialized as sparse matrices by the *Scipy* library [**?**] and the Table 2 shows the average processing time for global matrices assembly using gaussian quadrature with three gauss points in several unstructured linear triangular meshes. Whereas, the Table 3 shows the average processing time for assembly using the analytical elementary matrices. As can be seen, the global matrices assembly using analytical elementary matrices is about three times faster than the gaussian quadrature. However, the gaussian quadrature can be used on quadratic or cubic elements, where calculating the analytical elementary matrices is more costly.

Nodes	Elements	AVG Processing Time (s)
3436	6670	18.4
13745	26548	70.6
53229	105656	284.3
148187	295036	815.6

Table 2: Average processing time for global matrices assembly using gaussian quadrature in several unstrutured linear triangular meshes

Nodes	Elements	AVG Processing Time (s)
3436	6670	4.8
13745	26548	22.9
53229	105656	98.1
148187	295036	264.8

Table 3: Average processing time for global matrices assembly using analytical elementary matrices in several unstrutured linear triangular meshes

Then, the boundary conditions are applied by *ApplyBC* class that contains the boundary conditions for each benchmark problems of this work, namely: *Couette Flow, Poiseuille Flow, Half Poiseuille Flow, Lid-driven Cavity Flow, Backward-facing Step Flow, Pulsation Boundary Flow* and *Drug-eluting Stent Problems*. This class is enabled to apply the linear, quadrature and cubic boundary elements. The Table 4 shows the average processing time for the *Dirichlet* condition apply in several unstructured linear triangular meshes.

Nodes	Elements	AVG Processing Time (s)
3436	6670	1.4
13745	26548	6.7
53229	105656	26.9
148187	295036	81.8

Table 4: Average processing time for Dirichlet condition in several unstrutured linear triangular meshes

After boundary condition applied, the time loop is started and the updating mesh is done by the *MeshUpdate* class. In this class, the mesh velocity is calculated by the linear combination of Lagrangian and Laplacian smoothing velocities. Subsequently, the mesh is moved and the coordinate vectors and the global matrices must be reassembled, in addition to the boundary conditions reapplied. The Table 5 shows the average processing time for the *mesh update procedure* in several unstructured linear triangular meshes. This processing time does not take into account the global matrices assembly and boundary conditions apply (they are performed by Assembly and ApplyBC as previously mentioned), only the calculate mesh velocity and nodes mesh moving are considered. As can be seen, the processing time increases fourteen times while the mesh changes from 105656 to 295036 elements (about three times).

Nodes	Elements	AVG Processing Time (s)
3436	6670	0.5
13745	26548	2.8
53229	105656	26.8
148187	295036	384.4

Table 5: Average processing time for mesh update procedure in several unstrutured linear triangular meshes

Before solving the linear system, the semi-Lagrangian Method is applied by *Semi-Lagrangian* class. In this class, the searching procedure of the departure point and the interpolation of the ω_d^n and e_d^n is done. This class is enabled for the linear, quadratic and cubic triangular elements. The Table 6 shows the average processing time for the *semi-Lagragian* Method.

Nodes	Elements	AVG Processing Time (s)
3436	6670	0.6
13745	26548	2.1
53229	105656	8.5
148187	295036	24.9

Table 6: Average processing time for the semi-Lagrangian method in several unstrutured linear triangular meshes

Finally, the Vorticity-Streamfunction with species transport solver is done. As previously mentioned, the boundary condition of the vorticity is calculated for each time step and the linear equations system is solver by Scipy Conjugate Gradient and the end of time step, the convergence and steady state checks are done. The simulator is enabled to solver for linear, quadratic and cubic triangular elements. The Table 7 shows the average processing time for the *Vorticity Solver*.

Nodes	Elements	AVG Processing Time (s)
3436	6670	1.3
13745	26548	5.4
53229	105656	21.2
148187	295036	63.5

Table 7: Average processing time for Vorticity Solver in several unstrutured linear triangular meshes

Lastly, the parameters of simulation is printed and the VTK file is exported to perform the post-processing by *PARAVIEW* open source [**?**]. The Table 8 shows the average processing time for the *VTK file export* in several unstructured linear triangular meshes.

Nodes	Elements	AVG Processing Time (s)
3436	6670	0.1
13745	26548	0.3
53229	105656	1.4
148187	295036	3.9

Table 8: Average processing time for VTK file export in several unstrutured linear triangular meshes

In addition, the Table 9 shows the average computational cost ratio of the numerical code process for several linear triangular elements. As can be seen, the assembly process is the highest computational cost and improvements are expected to improve the code performance.

Process	AVG Computational Cost (%)
Mesh import	1.24
Assembly	73.87
BC Apply	6.70
Mesh update	10.05
Semi-Lagrangian	2.27
Vorticity Solver	5.51
VTK export	0.36

Table 9: Average computational cost for several linear triangular elements.

Therefore, the Figure 12 shows schematically the procedure used in this numerical simulation.
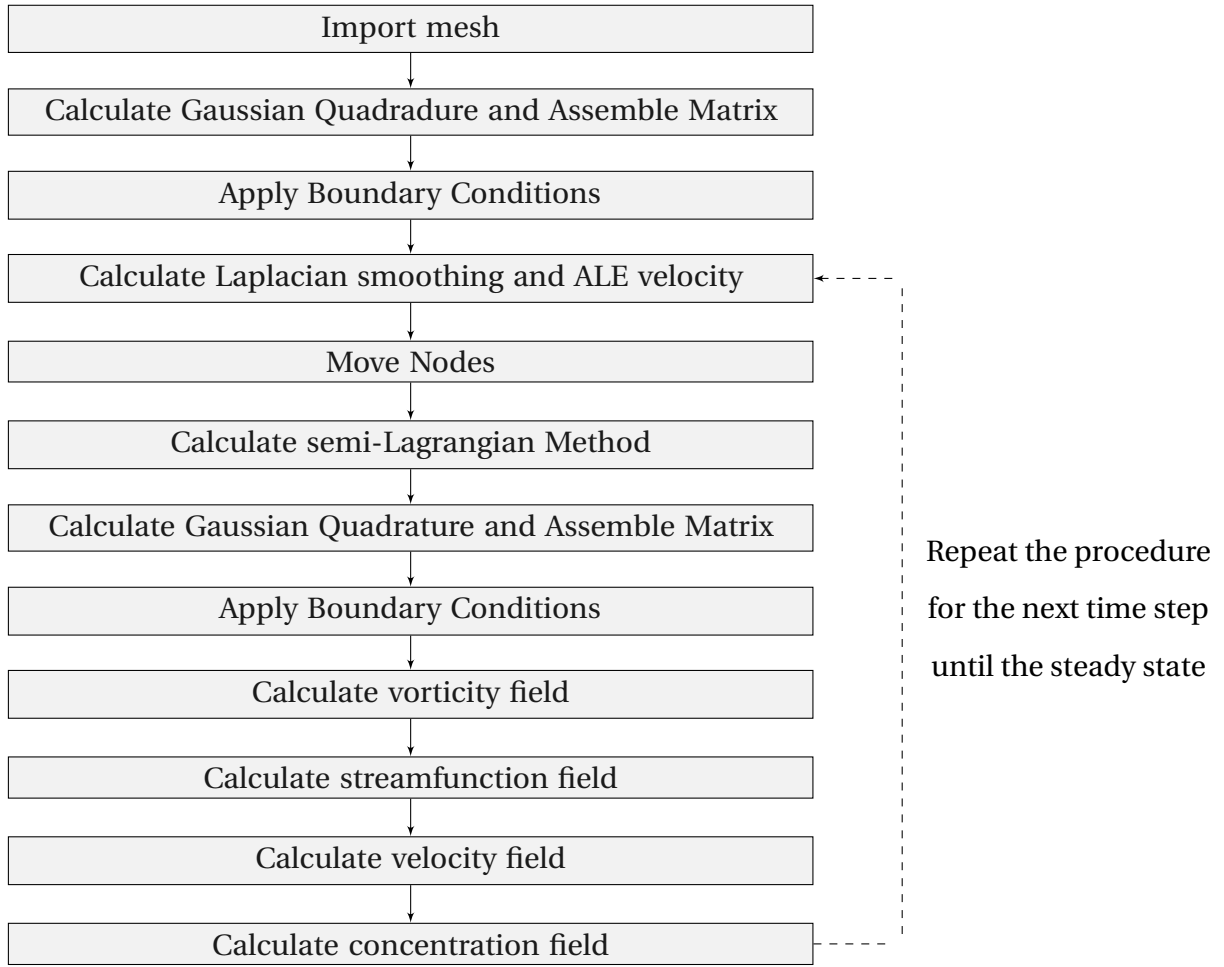
Figure 12: Solve algorithm for Vorticity-Streamfunction Formulation with Species Transport Equation using the semi-Lagrangian Method in an ALE context

4.3 **Laplacian Smoothing**

As previously mentioned, the computational mesh velocity assumes different values of velocity from the Eulerian and Lagrangean description. Thus, it can be represented as a linear combination of other velocities, such as:

$$\hat{\mathbf{v}} = \beta_1 \mathbf{v}_1 + \beta_2 \mathbf{v}_2 \qquad (4.1)$$

where, $\mathbf{v}_1$ is the Lagrangian velocity, $\mathbf{v}_2$ is the Laplacian smoothing velocity, β_1 is a parameter controls the Lagragian motion and β_2 controls the intensity of velocity smoothing. The choice of these velocity fields and their parameters must be in order to improve the result of the

numerical simulation and to avoid the degradation of the computational elements, especially close to the domain boundary, where the elements are compressed making the simulation unstable.

Lagrangian velocity is the material flow velocity. This portion makes the nodes of the computational mesh move in the same direction and sense as the flow velocity. The intensity is proportional to the value of parameter β_1. Depending on the chosen value, the insertion and deletion of nodes are required.

The velocity of Laplacian smoothing is that nodes acquire due to their topological redistribution. Considering a node in a non-uniform mesh, it will be moved to the centroid of the 1-ring neighbors. Thereby, it is expected the smoothing procedure converges to a more uniform point distribution, as shown in Figure 13.

(a)   (b)
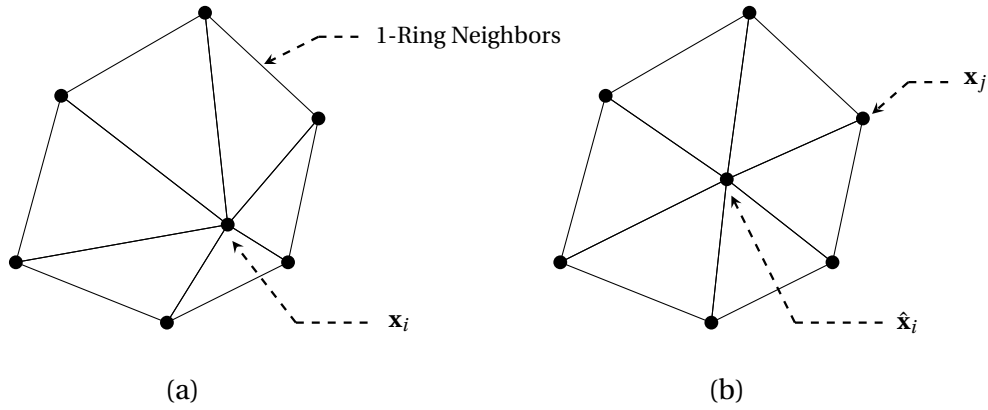
Figure 13: Laplacian smoothing in 2-dimensional space: (a) initial point position and (b) final point position after successively smoothing steps.

According to [?], the new node position $\hat{\mathbf{x}}_i$ can be approximated using an iterative weighted sum of the 1-ring neighbors of a node:

$$\hat{\mathbf{x}}_i = w_{ij} \sum_{j}^{N_1} \left(\mathbf{x}_j - \mathbf{x}_i \right) \tag{4.2}$$

where, w_{ij} is the weight and it can be calculated in several ways, N_1 is the 1-ring neighbors of a node, $\mathbf{x}_j$ is the coordinate vector of neighbor node and $\mathbf{x}_i$ is the coordinate vector of node that will be moved. It is possible to choose several strategies to calculate the weight of this

equation. In this work, the weight was calculated as the sum of the inverse distance from its neighbor vertices, that is:

$$w_{ij} = \sum_{j}^{N_1} \frac{1}{\left(\mathbf{x}_j - \mathbf{x}_i\right)} \tag{4.3}$$

Then, the Laplacian smoothing velocity can be calculated as:

$$\mathbf{v}_2 = w_{ij} \sum_{j}^{N_1} \frac{\left(\mathbf{x}_j - \mathbf{x}_i\right)}{\Delta t} \tag{4.4}$$

In Figure 14 is shown the Laplacian smoothing application in a moving boundary problem when it tends to the maximum compression state. In both simulation, was not considered the β_1. As can be seen, the elements close to the boundary are compressed until they leave the domain (Figure 14a). In this case, the Laplacian smoothing velocity was not considered ($\beta_2 = 0$). However, this effect is not seen when Laplacian smoothing is applied (Figure 14b). In this case, the β_2 parameter was considered assuming the maximum value, that is, $\beta_2 = 1$. In these cases, Laplacian smoothing becomes essential to perform the numerical simulation.

(a)   (b)
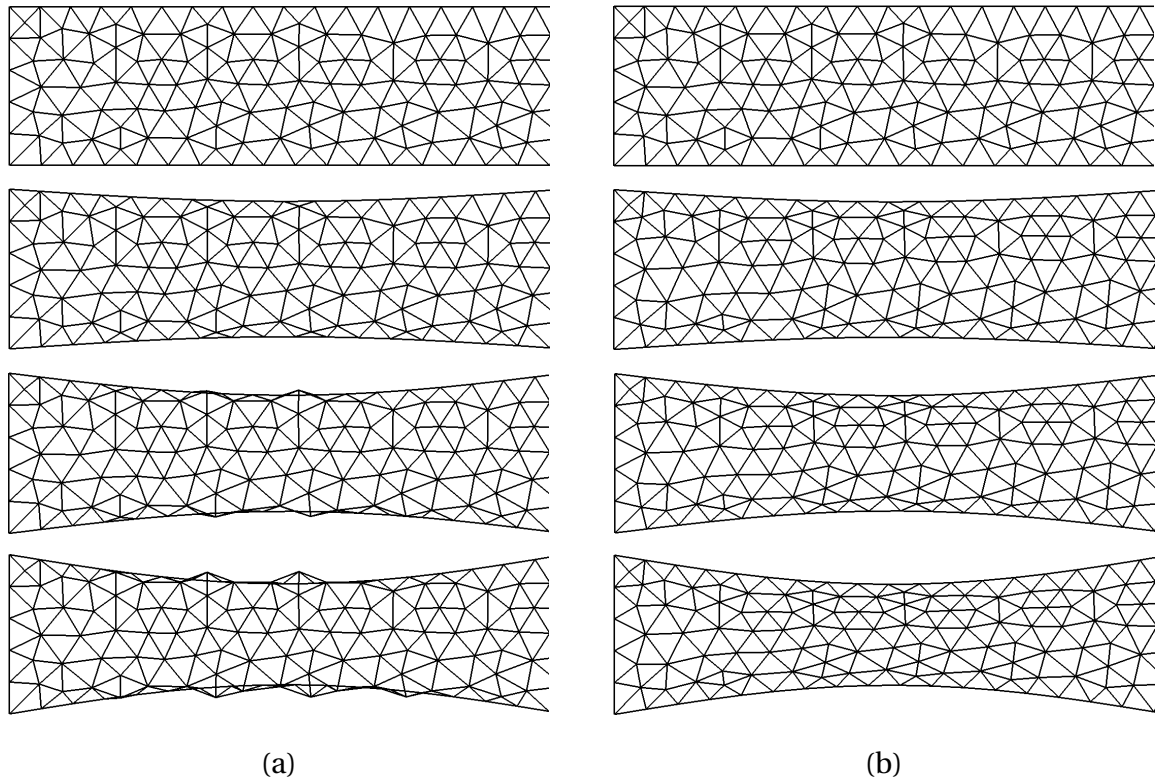
Figure 14: Note here that the boundary motion spoils the mesh structure in (a), thus a mesh treatment to avoid element collapse is required as in (b). The mesh treatment used in this work was the Laplacian Smoothing.

5 **VALIDATIONS**

5.1 **Introduction**

In this chapter, we will present the results obtained from four cases with the numerical simulation of the Navier Stokes equation using the vorticity-streamfunction formulation with the species transport equation, where we have incompressible and monophase two-dimensional flow for all cases in an Arbitrary Lagrangian-Eulerian context using the semi-Lagrangian Method. In all simulations, were considered the $\beta_1 = 0$ and $\beta_2 = 1$. Therefore, the computational mesh velocity is calculated using only the Laplacian smoothing velocity.

The first section is about *Couette flow* and the numerical solution is compared with the analytical solution. The second section is about the *Poiseuille flow*, where the no-slip condition is applied and the numerical solution is also compared with the analytical solution. The third section refers to the flow of *Poiseuille* in the half domain, where the free slip condition is applied on the axis of symmetry. The fourth section refers to the (*lid-driven cavity flow*) and the solution is compared with the results presented by Ghia et al. (1982) [?] and Marchi et al. (2009) [?] for several Reynold numbers. Lastly, the comparison of the semi-Lagrangian Method in a unstructured linear and quadratic triangular mesh is done for the transport of a scalar submitted to a pure advection flow is presented.

All numerical simulations were performed on the computers of *Numerical Simulations Laboratory (LEN)* of *Environmental Simulations in Reservoirs and Study Group (GESAR)* with the following configuration:

- AMD FX-8350 4GHz with 8 core, 32Gb RAM Memory, 1000Gb of HD. LINUX Ubuntu 16.04 LTS. The numerical code implementarion was performed using Python 2.7 language

5.2 **Couette Flow**

A monophase, steady and fully developed flow of a Newtonian and incompressible fluid between parallel horizontal plates, where the lower plate moves with U_{bottom} velocity and the upper plate moves with U_{top}, is known as *Couette flow*. The Figure 15 presents schematically this flow and the profile of the expected velocity field.
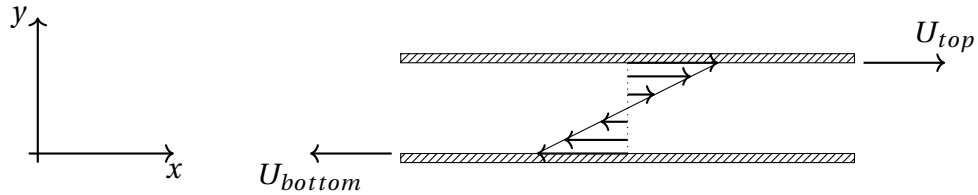

Figure 15: Couette Flow

The velocity profile equation is shown below:

$$u = \left[U_{top} - U_{bottom}\right]\frac{y}{L} + U_{bottom} \tag{5.1}$$

where U_{top} is the top plate velocity and its value is $U_{top} = 1$, U_{bottom} is the bottom plate velocity and its value is $U_{bottom} = -1$, L is non-dimensional length between the plates and its value is $L = 1$ and y is the vertical coordinates and it varies between $y = [0,1]$. The domaian was discretizes using a linear triangula mesh with 3835 nodes and 7299 elements.

The Figure 16 shows the steady state velocity profile when $Re = 100$. It is possible to observe that the numerical solution converges to the analytical solution.
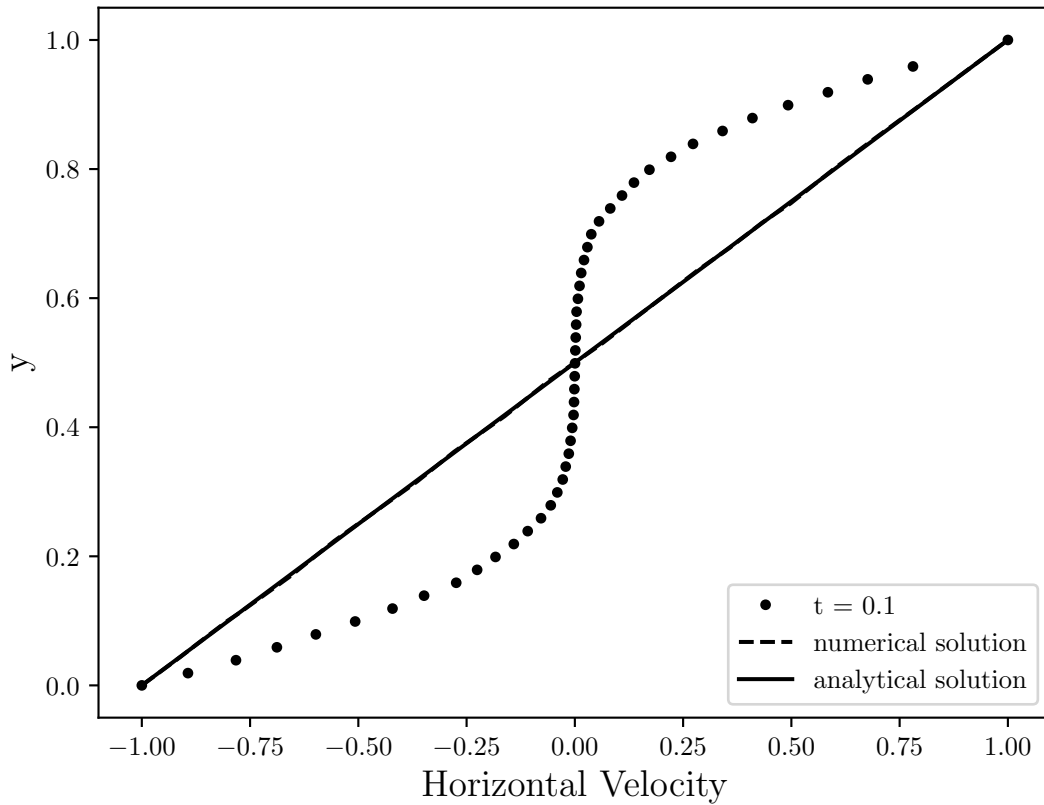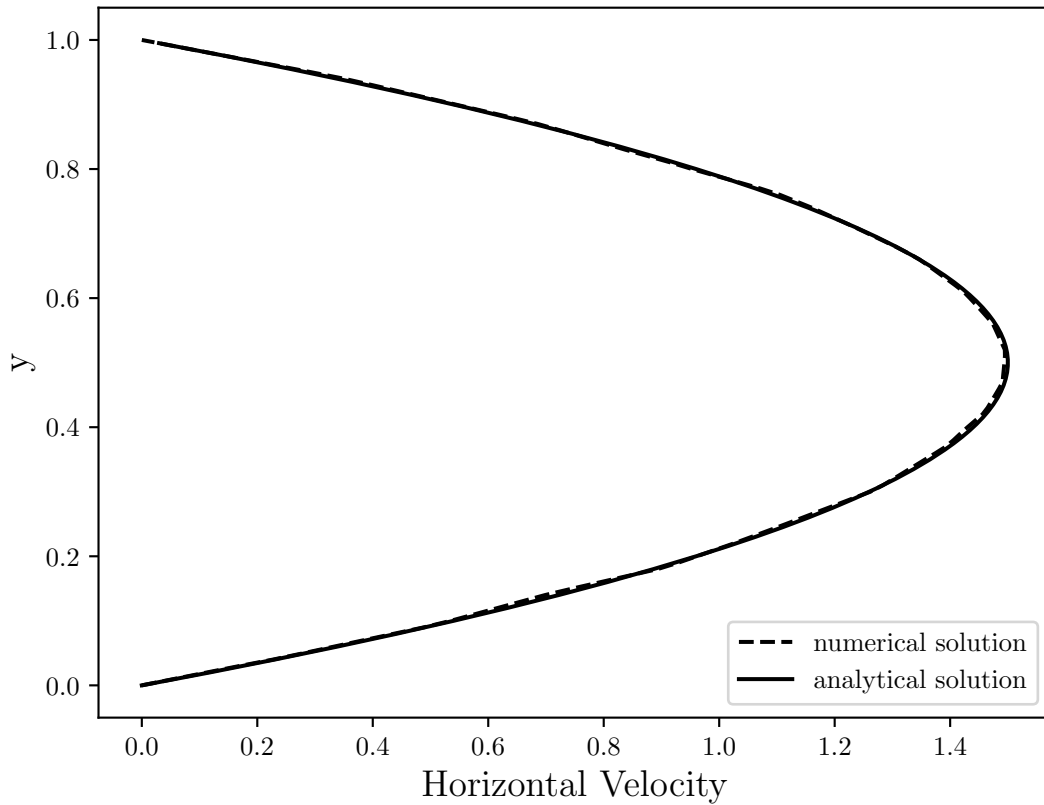
Figure 16: Steady state velocity profile when *Re* = 100 and the comparison between the numerical and analytical solution for Couette flow.

5.3 **Poiseuille Flow**

A monophase, steady and fully developed flow of a Newtonian and incompressible fluid between parallel and fixed horizontal plates is maintained due to a pressure gradient $\partial p/\partial x$ imposed as mentioned by Pontes and Mangiavacchi (2016) [**?**]. This flow is known as *Poiseuille flow.* The Figure 17 presents schematically this flow and the expected velocity field.
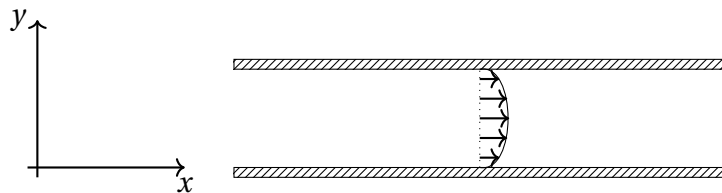
Figure 17: Poiseuille Flow

The velocity profile equation is shown below:

$$u = \frac{4u_{max}}{L^2}y\big[L-y\big] \tag{5.2}$$

where u_{max} is maximum velocity and its value is $u_{max} = 1.5$, L is non-dimensional length between the plates and its value is $L = 1$ and y is the vertical coordinates and it varies between $y = [0,1]$. The domain was discretized using a linear triangular mesh with 3835 nodes and 7299 elements.

The Figure 18 shows the steady state velocity profile when $Re = 100$. It is possible to observe that the numerical solution converges to the analytical solution.

Figure 18: Steady state velocity profile when $Re = 100$ and the comparison between the numerical and analytical solution for Poiseuille flow.

The Table 10 shows the relative error between the numerical solution and the analytical solution for several unstructured linear triangular meshes, ranging from 100 to 25600 linear triangular elements. For the mesh with 7299 elements as in the case of this benchmark problem, the estimated relative error for the velocity field is 0.4%.

Elements	Error (%)
100	25.00
400	7.47
1600	2.11
6400	0.61
25600	0.17

Table 10: The relative error of numerical solution for several elements numbers in an unstructured linear triangular mesh.

The relative error was estimated as:

$$Error = \sqrt{\frac{\sum (v_n - v_a)^2}{\sum |v_a|^2}} \tag{5.3}$$

where v_n is the numerical velocity fielf and v_a is the analytical velocity field.

The Figure 19 presents the relative error of the numerical solution with the first and second order convergence curves on a log-log scale. As can be seen, the relative error of the numerical solution for Poiseuille flow has the form of first order convergence. Thus, when increasing the number of elements, the relative error of the numerical solution regresses linearly. A possible cause is due to the vorticity boundary condition used in this work. Therefore, the use of the 2a-order boundary condition would possibly improve the order convergence of the numerical solution. However, it is necessary to be investigated.
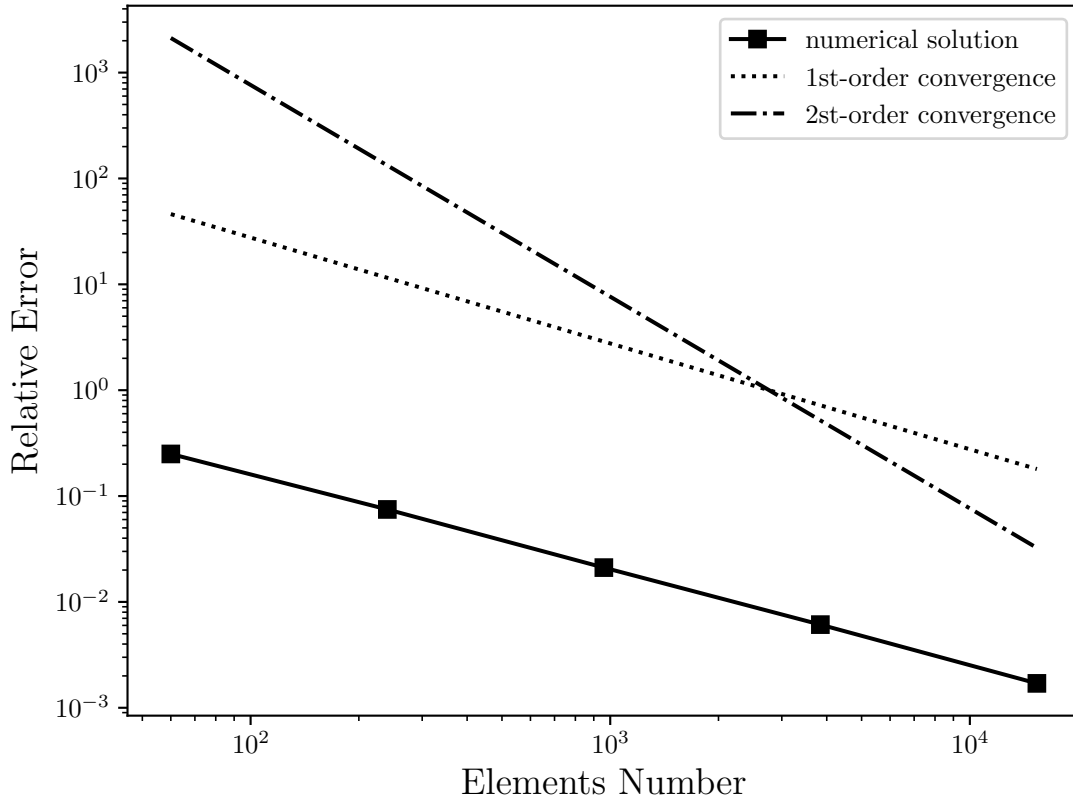
Figure 19: Convergence order in log-log scale: It is estimated the the relative error of numerical solution has first order convergence.

5.4 **Half Poiseuille Flow**

This section presents the simulation of the *Poiseuille* flow in half of the domain. Thus, the free-slip condition is required on the axis of symmetry. The Figure 20 presents schematically this flow with the specified axis of symmetry and the expected velocity field.
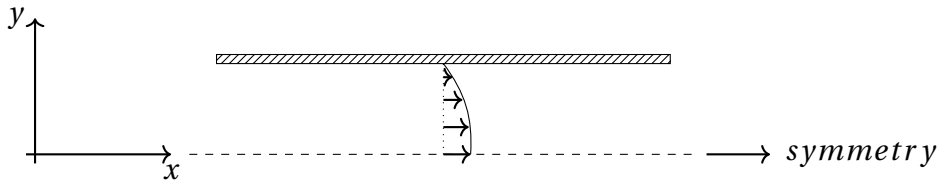

Figure 20: Half Poiseuille flow

The velocity profile equation is shown below:

$$u = u_{max}\left[1 - \frac{y^2}{L^2}\right]$$ (5.4)

where u_{max} is maximum velocity and its value is $u_{max} = 1.5$, L is non-dimensional length between the plates and its value is $L = 1$ and y is the vertical coordinates and it varies between $y = [0,1]$. The domain was discretized using a linear triangular mesh with 3835 nodes and 7299 elements.

The Figure 21 shows the unsteady velocity profile when $Re = 100$, in addition to the comparison between the numerical and analytical solutions in the steady state of proposed problem. It is possible to observe that the numerical solution converges to the analytical solution when the flow becomes steady state.
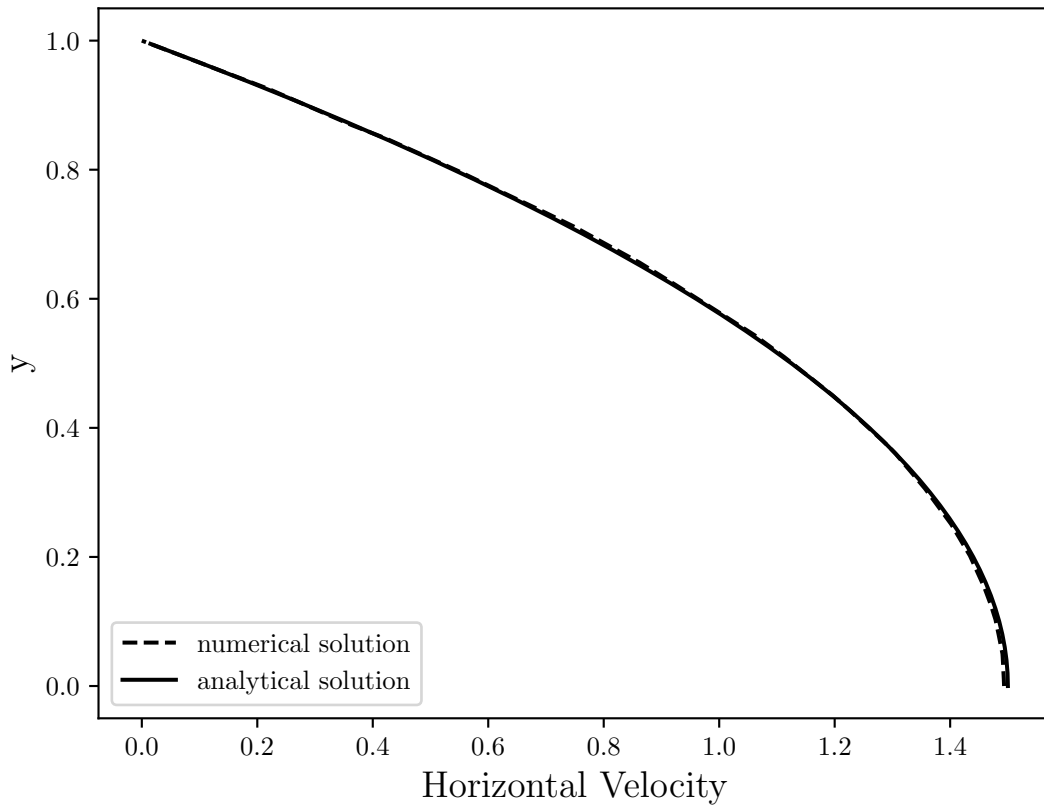
Figure 21: Unsteady velocity profile when $Re = 100$ and the comparison between the numerical and analytical solution for Half Poiseuille flow.

5.5 **Lid-Driven Cavity**

A flow in a cavity where the side and bottom walls are fixes and the cover moves at a constant velocity such as $U_{top} = 1$ is known as *Lid-driven Cavity flow*. In addition to yhe streamfunction is set null value in all boundary, because there is no volumetric flux crossing the boundaries in lid-driven cavity flow. The Figure 22 presents schematically this flow and the expected velocity field.
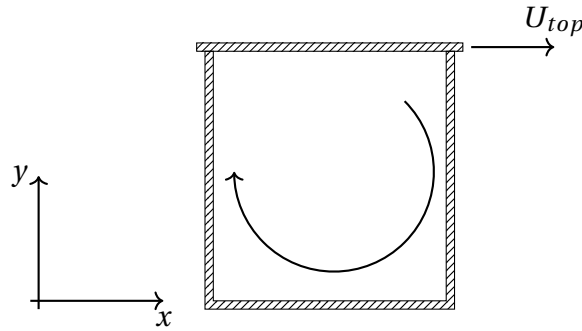

Figure 22: Lid-driven Cavity Flow

The benchmark problem were simulated with the following Reynolds numbers (*Re*): 10, 100, 400 and 1000. The Figure 23 and Figure 24 present the profile of the horizontal and vertical velocities, respectively, for steady state in several Reynolds number. They were compared with Ghia et al. (1982) [?] and Marchi et al. (2009) [?]. The domain was discretized using a linear triangular mesh with 1563 nodes and 2988 elements.
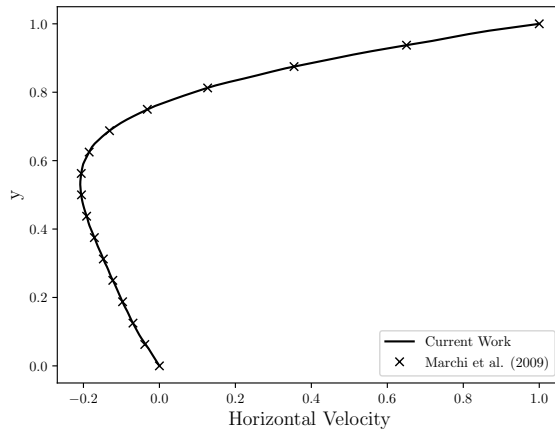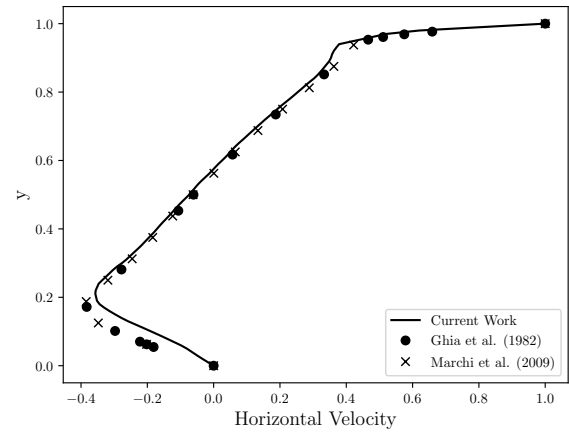
(a)
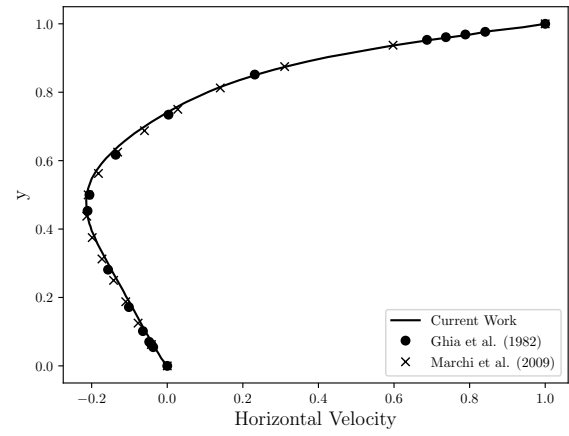
(b)
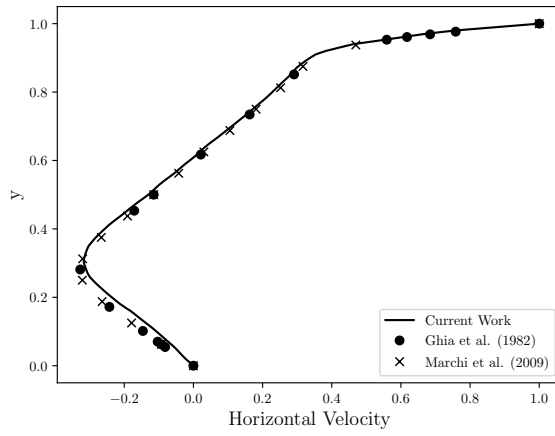
(c)

(d)

Figure 23: The horizontal velocity profile in central line of cavity ($x = 0.5$) for several Reynolds number: (a) 10 (b) 100 (c) 400 (d) 1000.
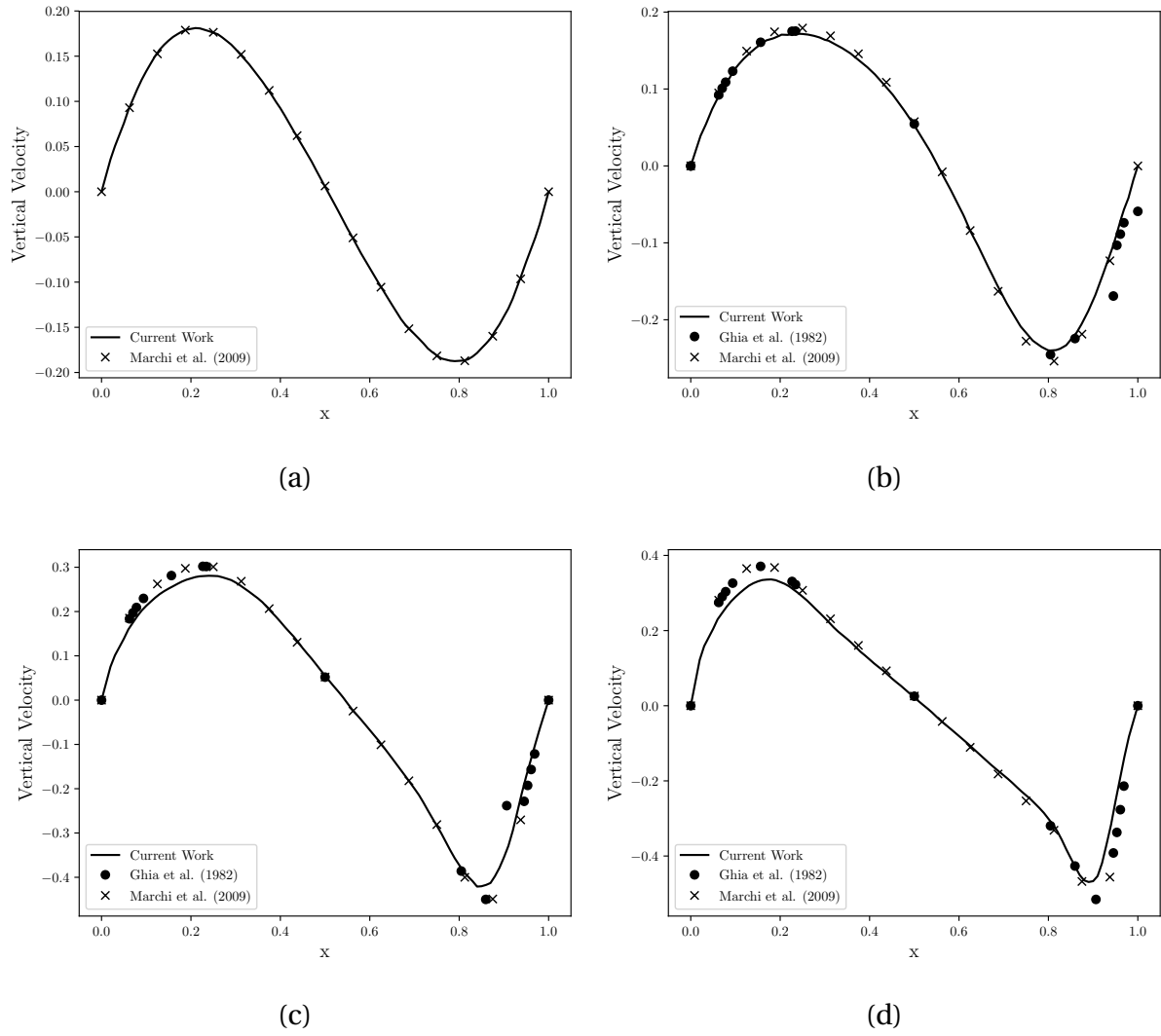
Figure 24: The vertical velocity profile in central line of cavity ($y = 0.5$) for several Reynolds number: (a) 10 (b) 100 (c) 400 (d) 1000.

5.6 **Pure Advection Flow**

The transport of a scalar according to a parabolic function and submitted to a monophase flow of a Newtonian and incompressible fluid with a high number of *Reynolds* ($Re \rightarrow \infty$) is known as a *Pure Advection flow*. In this type of flow, it is expected that the scalar will not diffuse. For approximation methods like *FEM* and *FDM*, it is possible to observe the presence of spurious oscillations. As mentioned earlier, several schemes can be used to reduce these numerical oscillations. In this section, we will present the use of the *semi-Lagrangian* method to reduce spurious oscillations. In addition, it will be compared to the analytical solution for an unstructured linear and quadratic triangular mesh in order to analyze the

numerical diffusion due to the interpolation as mentioned in the section 3.5. The Figure 25 presents schematically the problem and the dynamics of scalar transport.
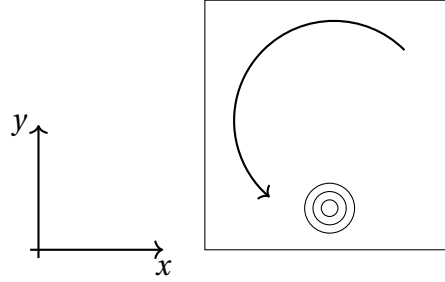

Figure 25: Transport of a scalar in Pure Advection Flow.

The scalar transport α for a pure advection flow is shown below:

$$\frac{\partial \alpha}{\partial t} + \mathbf{v} \cdot \nabla \alpha = 0 \tag{5.5}$$

where $\mathbf{v} = (u, v)$ is velocity field and its components are defined as: $u = -y$ e $v = x$. Therefore, it is expected that given an initial scalar field, it will be displaced by the velocity field without diffusion, that is, its profile should not be changed while the flow occurs. Any change in the scalar field profile is considered a numerical error. For this numerical simulation, the domain was discretized using an unstructured linear triangular mesh with 2216 nodes and 4270 elements and an unstructured quadratic triangular mesh with 8701 nodes and 4270 elements. In addition to the initial scalar field was defined by a parabolic profile $c = 1 - x^2 - y^2$, where x and y are space components.

The Figure 26 shows the comparison between the scalar field profile α for the numerical solution using the semi-Lagrangian Method in an unstructured linear triangular mesh and the analytical solution in several positions on the axis of rotation as the flow occurs. As can be seen, the spurious oscillations are not observed, however the scalar field profile in semi-Lagrangian Method is diffused due to the interpolation. Therefore, for this problem, the semi-Lagrangian Method in an unstructured linear triangular mesh did not present a satisfactory result.
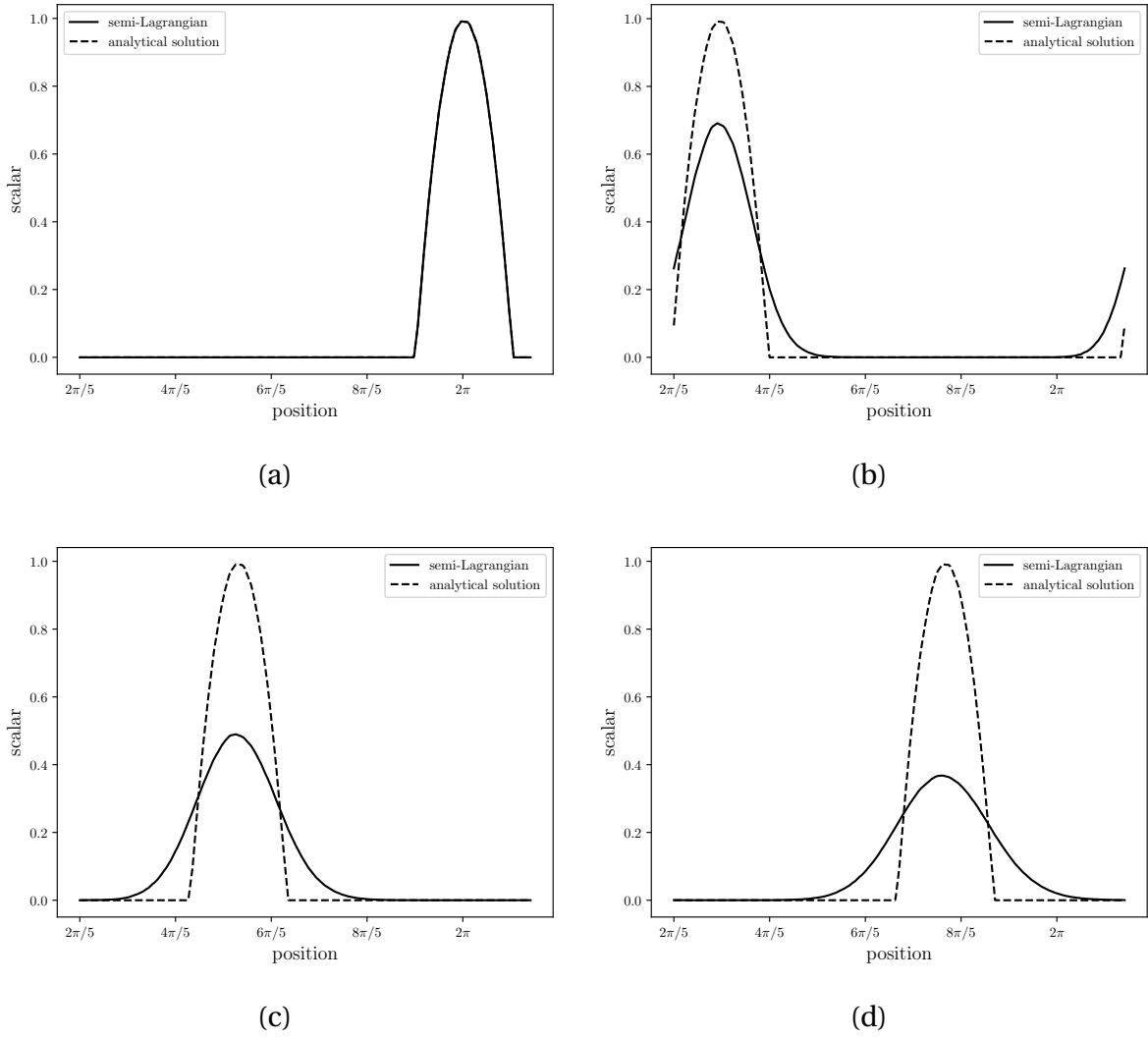
Figure 26: Comparison of α profile for the semi-Lagrangian Method and analytical solution in several positions on the axis of rotation using an unstructured linear triangular mesh: (a) initial, (b) 1/4 rotation, (c) 1/2 rotation and (d) 3/4 rotation.

The Figure 27 shows the comparison between the semi-Lagrangian and analytical solution for the same previously scalar field profile α, however an unstructured quadratic triangular mesh is used. As can be seen, the scalar field profile in semi-Lagrangian Method shows low numerical diffusion and the numerical result is better than the previous one. In addition, the spurious oscillations are not observed. Therefore, for this problem, the semi-Lagrangian Method in an unstructured quadratic triangular mesh present a satisfactory result.
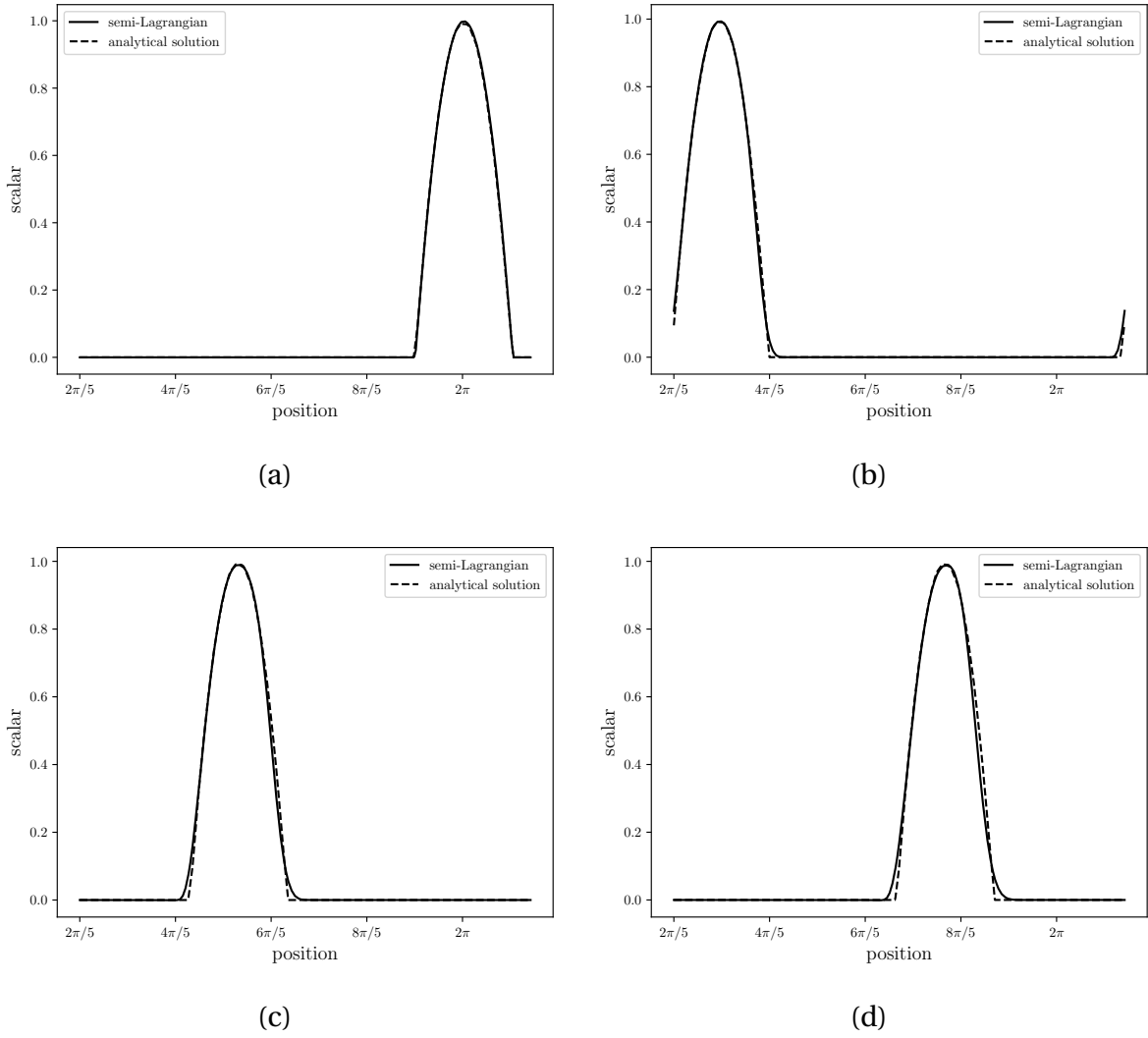
Figure 27: Comparison of α profile for the semi-Lagrangian Method and analytical solution in several positions on the axis of rotation using an unstrutured quadratic triangular mesh: (a) initial, (b) 1/4 rotation, (c) 1/2 rotation and (d) 3/4 rotation.

The Figure 28 and Figure 29 show the spatial and temporal evolution of the scalar profile for the semi-Lagrangian Method in an unstructured linear and quadratic triangular mesh respectively. As mentioned earlier, the spurious oscillations are not presented in both simulations. However, in the Figure 28, it is possible to observe the numerical diffusion, making its use not recommended for high Reynolds number problem.
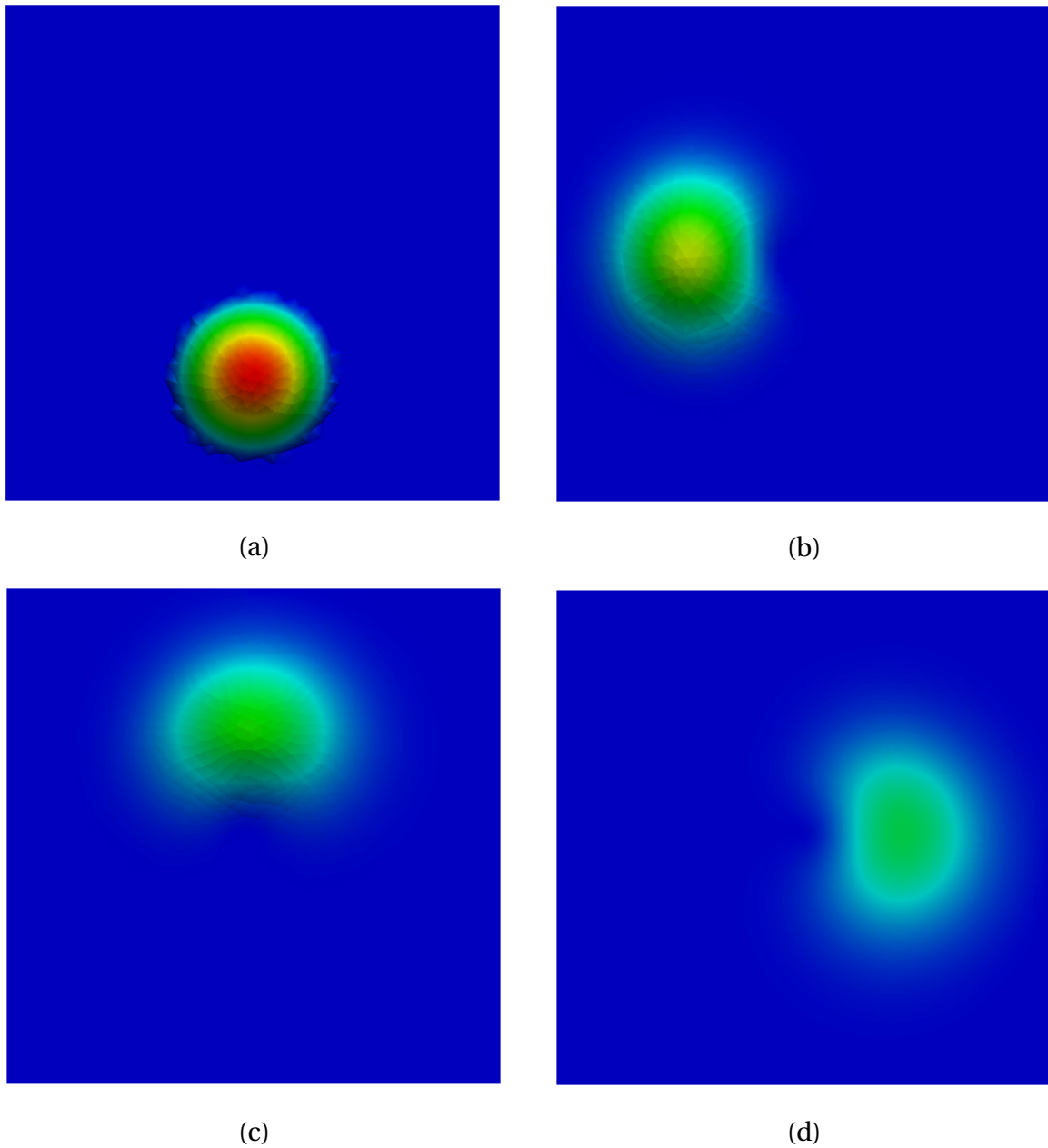
Figure 28: Spatial and temporal evolution for the semi-Lagrangian Method in several positions of the axis of rotation using an unstructured linear triangular mesh: (a) initial, (b) 1/4 rotation, (c) 1/2 rotation and (d) 3/4 rotation.
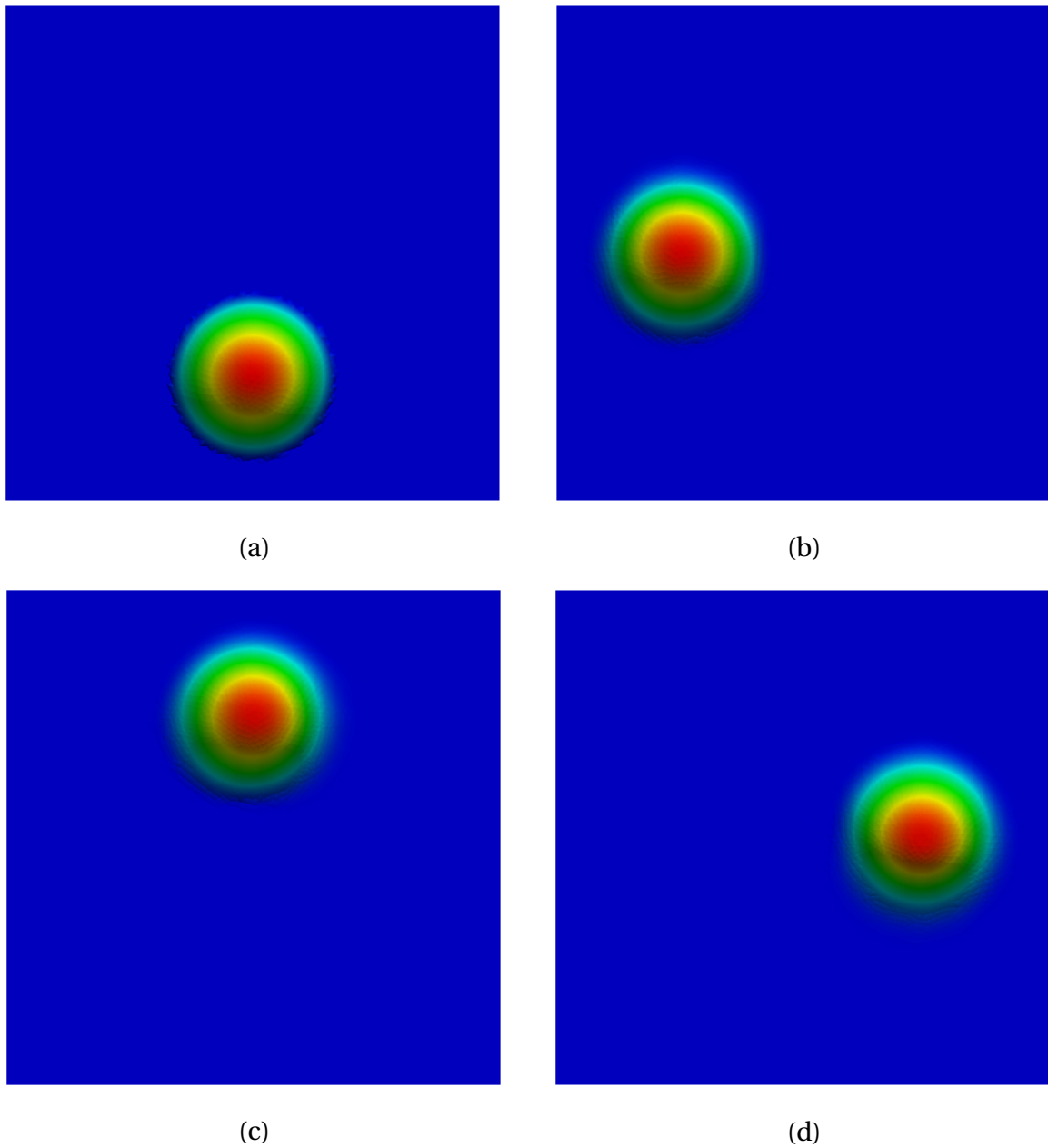
Figure 29: Spatial and temporal evolution for the semi-Lagrangian Method in several positions of the axis of rotation using an unstructured quadratic triangular mesh: (a) initial, (b) 1/4 rotation, (c) 1/2 rotation and (d) 3/4 rotation.

6 RESULTS

6.1 Introduction

In this chapter, the results of numerical simulations for blood flow in a coronary artery are presented. The lumen radius of the coronary artery used was $R = 0.0015m$, the viscosity used was $\mu = 0.0035Pa.s$ and the specific gravity used was $\rho = 1060kg/m^3$ as suggested by Bozsak, Chomaz and Barakat (2014) [?]. According to Kessler et al. (1998) [?], the blood velocity in the coronary artery is $u = 12cm/s$. Thus, the Reynolds number used will be $Re = 54.5$.

The Navier-Stokes equation is used according to the vorticity-streamfunction formulation with the species transport equation for two geometries proposed by Wang et al. (2017) [?], however modified to cartesian coordinates as shown in Figure 30. In the section 6.2, the numerical simulation for the coronary artery with atherosclerosis and a drug-eluting stent in a curved channel model is presented for several *Schmidt* number, such as 1, 10, 100 and 1000. In the 6.3, the real coronary artery with atherosclerosis and a drug-eluting stent is also simulated with several numbers of *Schmidt* number as in the previous section. Due to symmetry, only half of the domain is shown. The simulation was visualized using the *Paraview* open-source software proposed by Henderson (2007) [?].
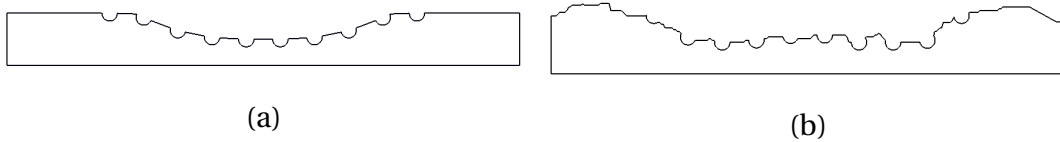

(a)   (b)

Figure 30: Non-dimensional domain for the blood flow in coronary artery The radius used was $R = 1$ and and the lumen length was $L = 10R$. (a) Curved Channel with Drug-Eluting Stent (b) Real Channel with Drug-Eluting Stent.

6.2 Curved Channel with Drug-Eluting Stent

In this section, will be present the case where the coronary artery has atherosclerosis and the drug-eluting stent is placed. It is modeled by 10 uniformly spaced semi-circles. The geometry used promotes a smooth reduction of the distance between the upper wall and symmetry axis of the channel. Due to atherosclerosis, 40% channel obstruction was considered and the domain was discretized using 15875 nodes and 35408 linear triangular elements.

The Figure 31 shows the unsteady state velocity profile in the middle section channel, that is, $x = 5.0R$. As expected, the numerical solution tends to a similar profile to the Half Poiseuille, as presented in the section 5.4. However, it is possible to observe an inversion of the velocity field sense at the top of the figure. This inversion occurs in the region that is located between the stents strut semi-circles. It is also possible to observe that the maximum horizontal velocity field in curvel channel reaches the $u = 3.43$ non-dimensional value, that is, more than 3 times the blood velocity in coronary artery without atherosclerosis and stent strut placed. This increase can influence the dynamics of blood flow and its biological processes and a more detailed analysis should be performed.
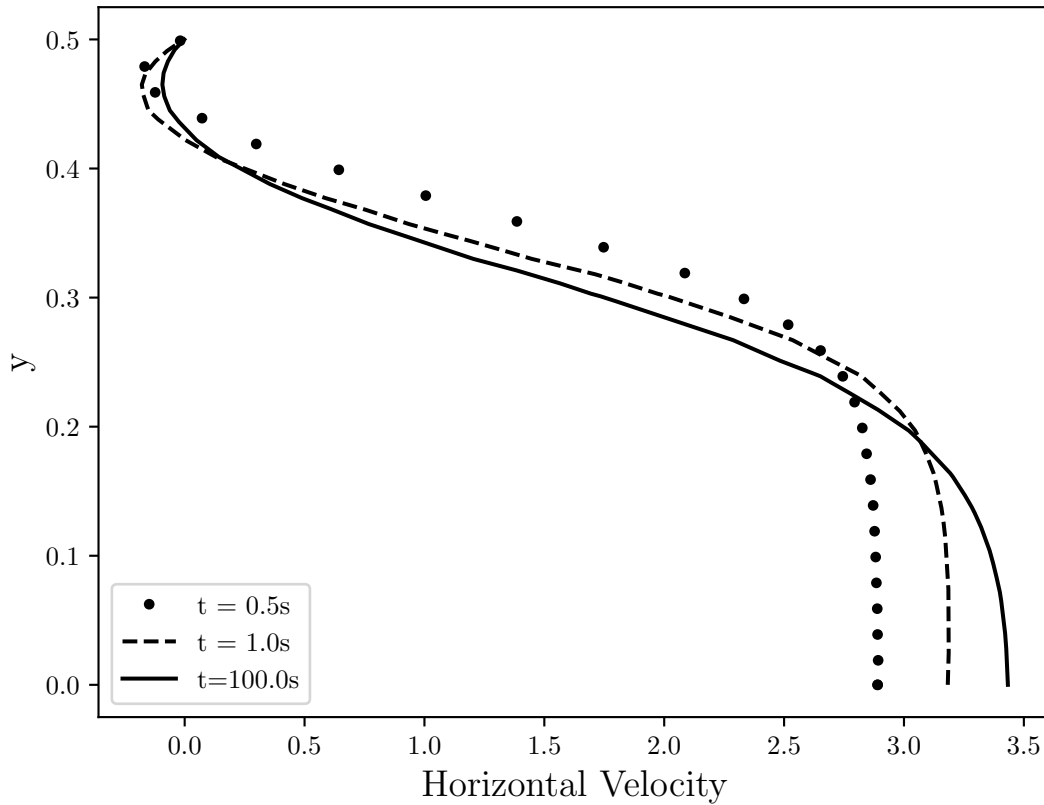

Figure 31: The unsteady state velocity profile in the middle ($x = 5.0R$) of the curved channel with drug-eluting stent.

The Figure 32 presents the evolution in time and space of the velocity field for half of the domain due to the symmetry of the solution. The velocity field is represented with non-dimensional values where the red color refers to the $u = 3.43$ value and the blue color $u = 0$ value. Conveting to dimensional values, we have $u = 41.16 cm/s$ and $u = 0 cm/s$ respectively. As can be seen, the region with the lowest horizontal velocity magnitude is found close to the boundary where the no-slip condition is applied. Whereas, it is increase close to the symmetric axis. Moreover, the largest horizontal velocity value is found in the maximum contraction region.
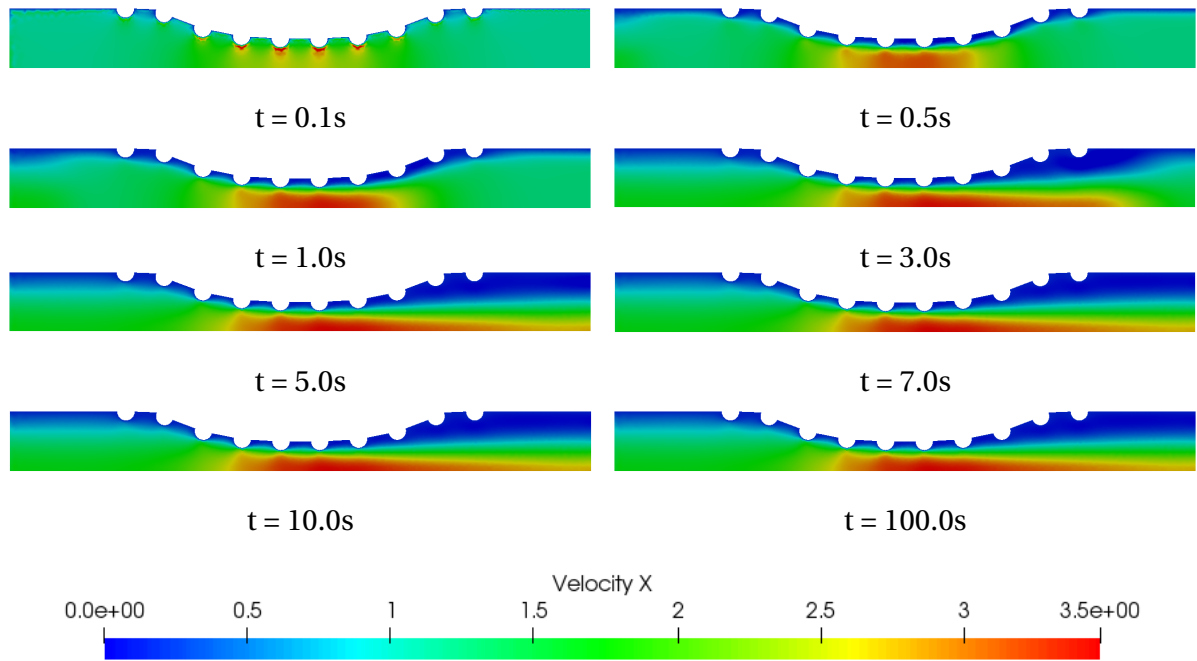

Figure 32: Temporal and spatial evolution of the velocity field for curved channel with drug-eluting stent.

As mentioned by Lucena et al. (2018) [**?**], it is estimated that 47% of the drug is diffused to the lumem and it is lost to the bloodstream. The Figure 33 to Figure 36 show the temporal and spatial evolution of the concentration field for several *Schmidt* number, such as: 1, 10, 100 and 1000 respectively. The concentration field is represented with the non-dimensional values where the red color represents 100% and the blue color represents 0% of the diffused concentration in the bloodstream.

It is possible to observe in the Figure 33 that the concentration field is significally more diffuse than in the Figure 34, where the Schmidt number is 10 times higher. This means that a large portion of the diffused drug in bloodstream is quickly spread in the $Sc = 1$ case. Moreover, in both cases, the concentration field is more dispersed at the end of the curved channel due to the sense of the blood flow. It is possible that the drug concentration diffused affects the density and viscosity of the blood and consequently the Reynolds number. Thefore, the velocity field would also be affected. However, this influence is not considered in this work.
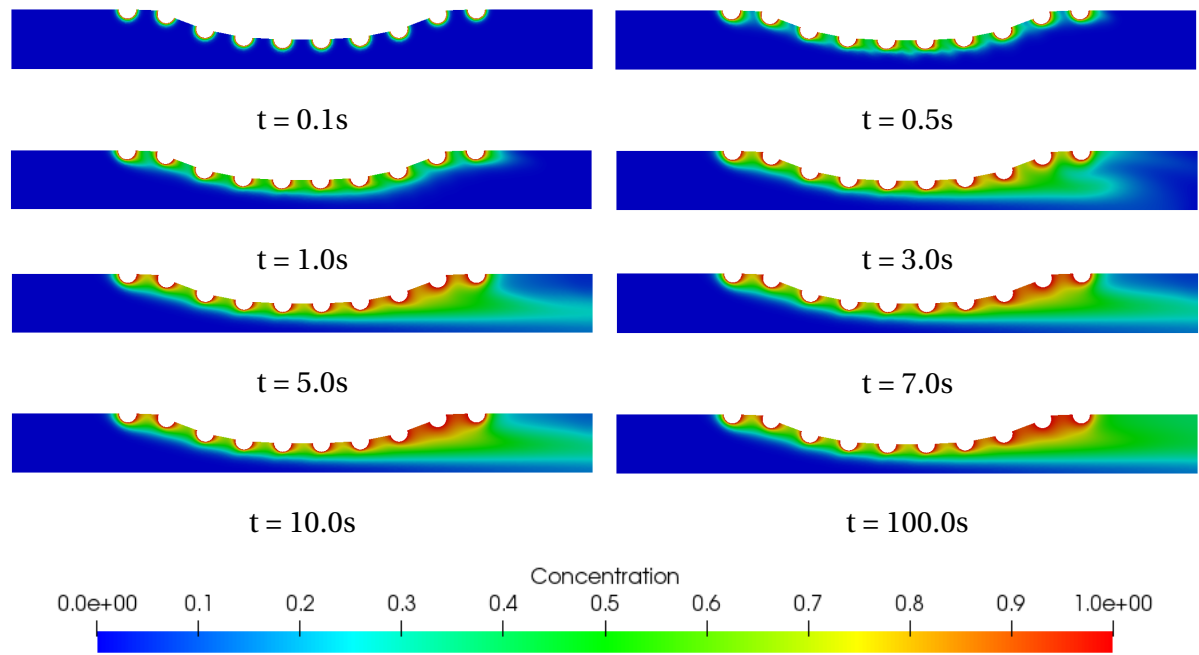

Figure 33: Temporal and spatial evolution of the concentration fiel for curved channel with drug-eluting stent and $Sc = 1$.

t = 0.1s

t = 0.5s

t = 1.0s

t = 3.0s

t = 5.0s

t = 7.0s

t = 10.0s

t = 100.0s

Concentration

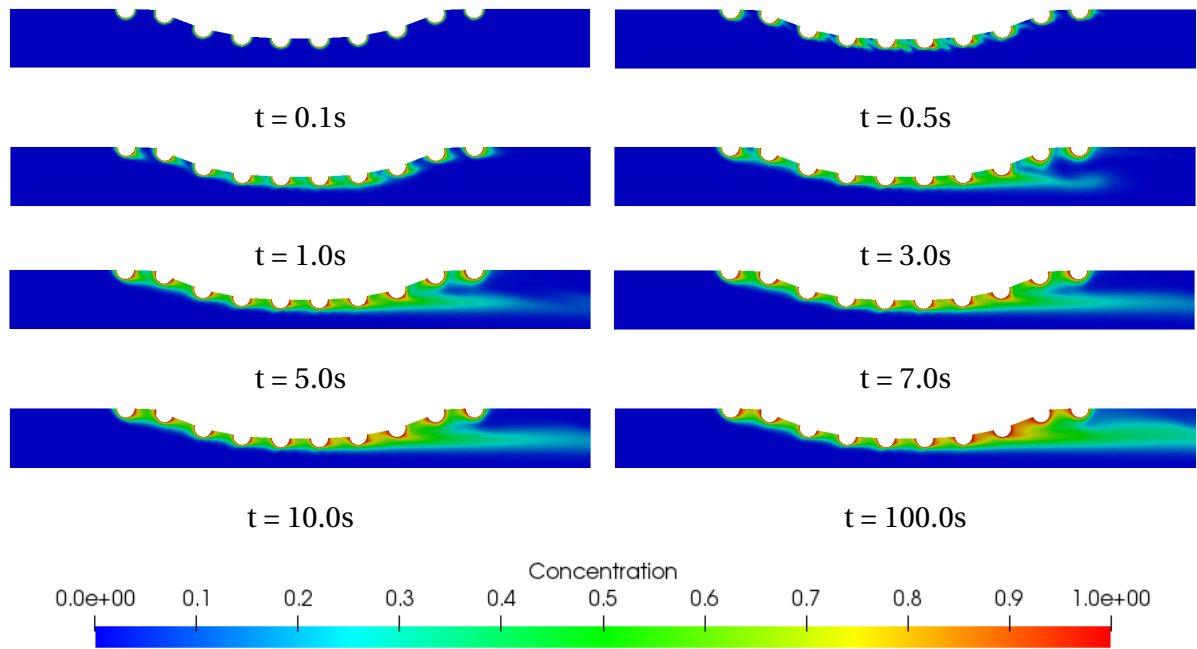0.0e+00   0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9   1.0e+00

Figure 34: Temporal and spatial evolution of the concentration fiel for curved channel with drug-eluting stent and $Sc = 10$.

For the Figure 35 and the Figure 36, the Schmidt number is increased to 100 and 1000 respectively. Both cases have a similar dynamic where it is possible to observe a decrease in the drug diffusion in the bloodstream when compared to the previous cases. Another important fact is the similarity in the concentration field for $Sc = 100$ and $Sc = 1000$. Therefore, it is necessary to simulate for longer times. In addition, it is possible to conclude that the *Schmidt* number directly influences the drug transport in the blood flow and for high values of the *Schmidt* number, the transport of chemical species becomes purely convective.
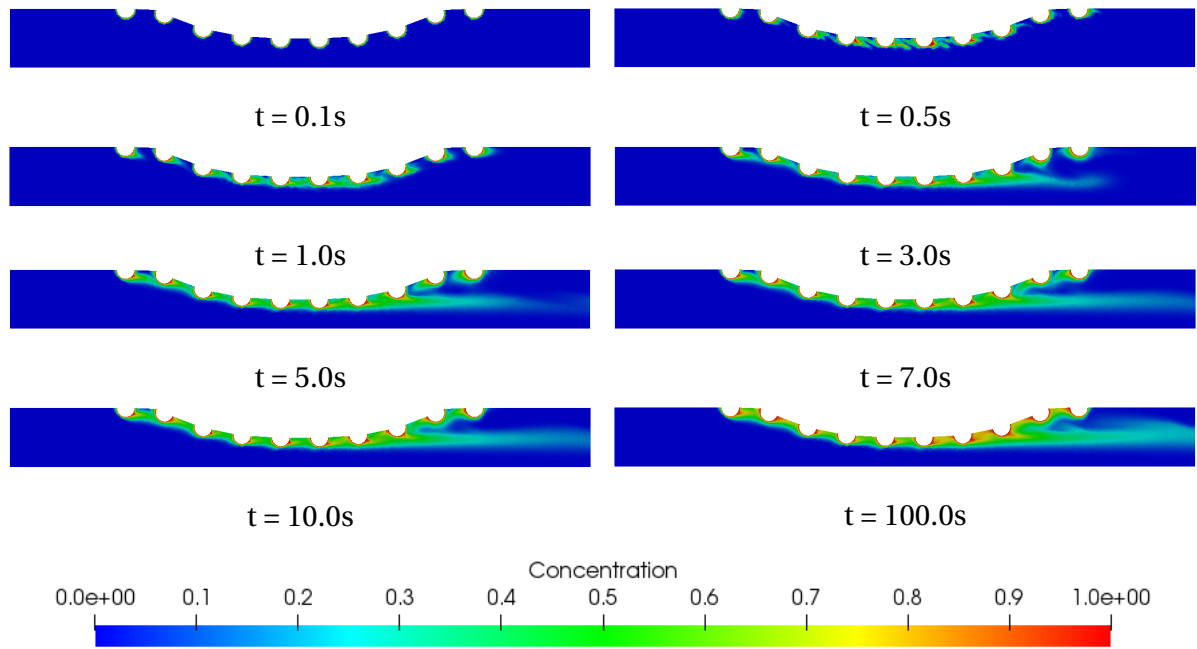
Figure 35: Temporal and spatial evolution of the concentration fiel for curved channel with drug-eluting stent and $Sc = 100$.
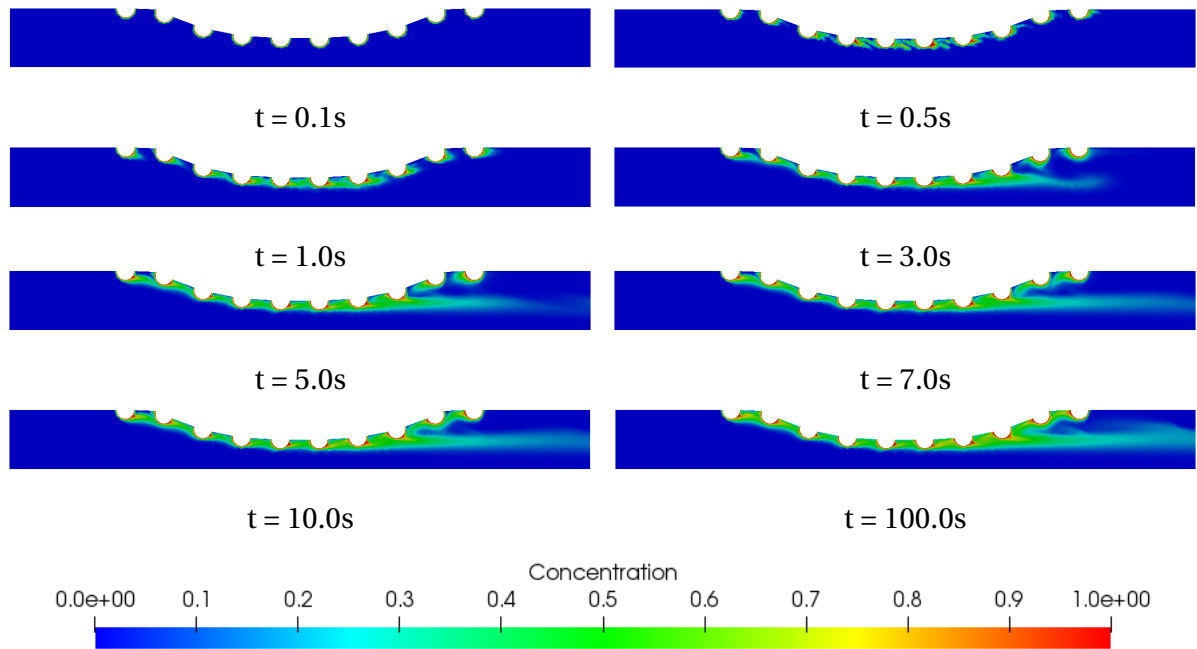

Figure 36: Temporal and spatial evolution of the concentration fiel for curved channel with drug-eluting stent and $Sc = 1000$.

6.3 Real Channel with Drug-Eluting Stent

For this case, the numerical simulation is performed for a real coronary artery with atherosclerosis whose geometry was obtained through an image processing as suggested by

Wang et al. (2017) [**?**]. This geometry is particular to each patient due to the patient health conditions. As in the previous case, the stent strut was modeled by 10 uniformly spaced semi-circles. The domain was discretized using 11807 nodes and 26426 linear triangular elements.

The Figure 37 shows the unsteady state velocity profile in the middle of the real channel, that is, $x = 5.0R$. Even so, the numerical solution shown a simular profile to the previous case, however it is not observed the inversion of the velocity field sense. It is also possible to observer that the maximum non-dimensional value of the velocity field reaches $u = 3.40$ close to symmetric axis. However, this velocity may vary according to the coronary artery geometry for each patient. It compared to the curved channel, the maximum horizontal velocity had a difference less than 1%. Therefore, the curved channel shown an acceptable approach, besides this model being simpler to implement and to perform the numerical simulations than real channel model.
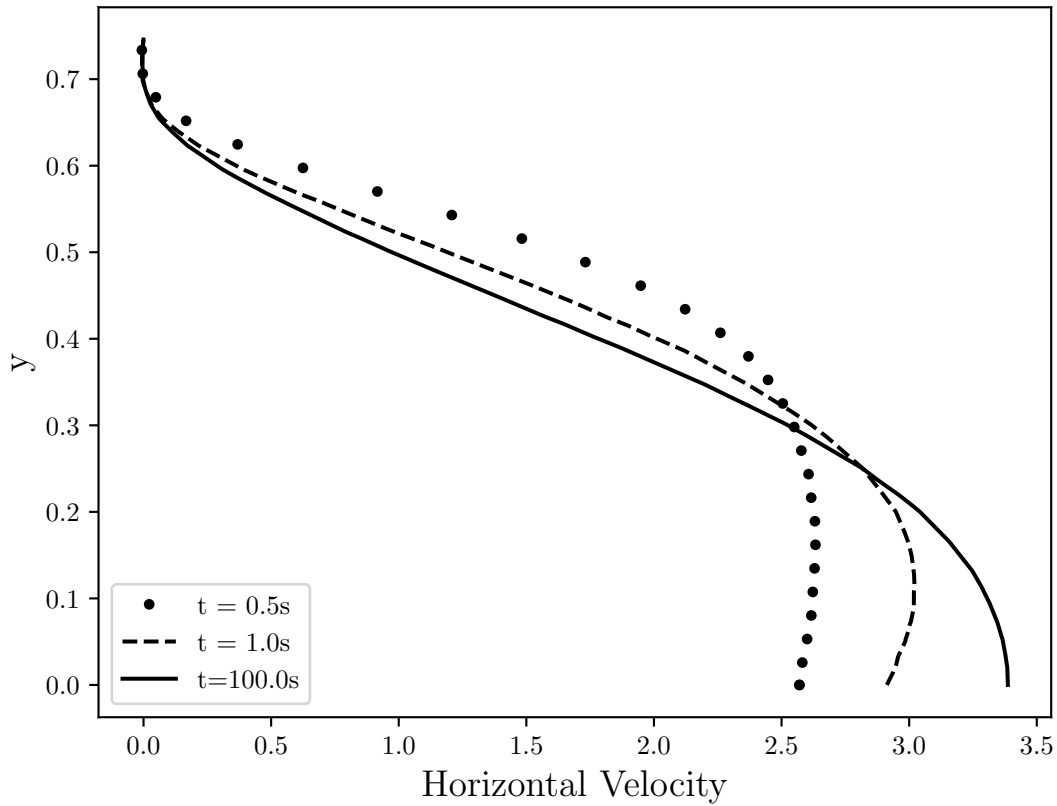

Figure 37: The unsteady state velocity profile in the middle of the real channel with drug-eluting stent.

The Figure 38 presents the evolution in time and space of the velocity field for half of the domain. The velocity field is represented with non-dimensional values where the red color refers to the $u = 3.40$ value and the blue color $u = 0$ value. Conveting to dimensional values, we have $u = 40.8 cm/s$ and $u = 0 cm/s$ respectively. As in the previous case, the region with the lowest horizontal velocity magnitue is found close to the boundary due to the no-slip condition and the largest horizontal velocity value is found close to symmetric axis. For this real channel, is possible to observe that there is two contraction region where the velocity field is increased. However, this coronary artery geometry varies for each patient and its blood dynamics can be changed.
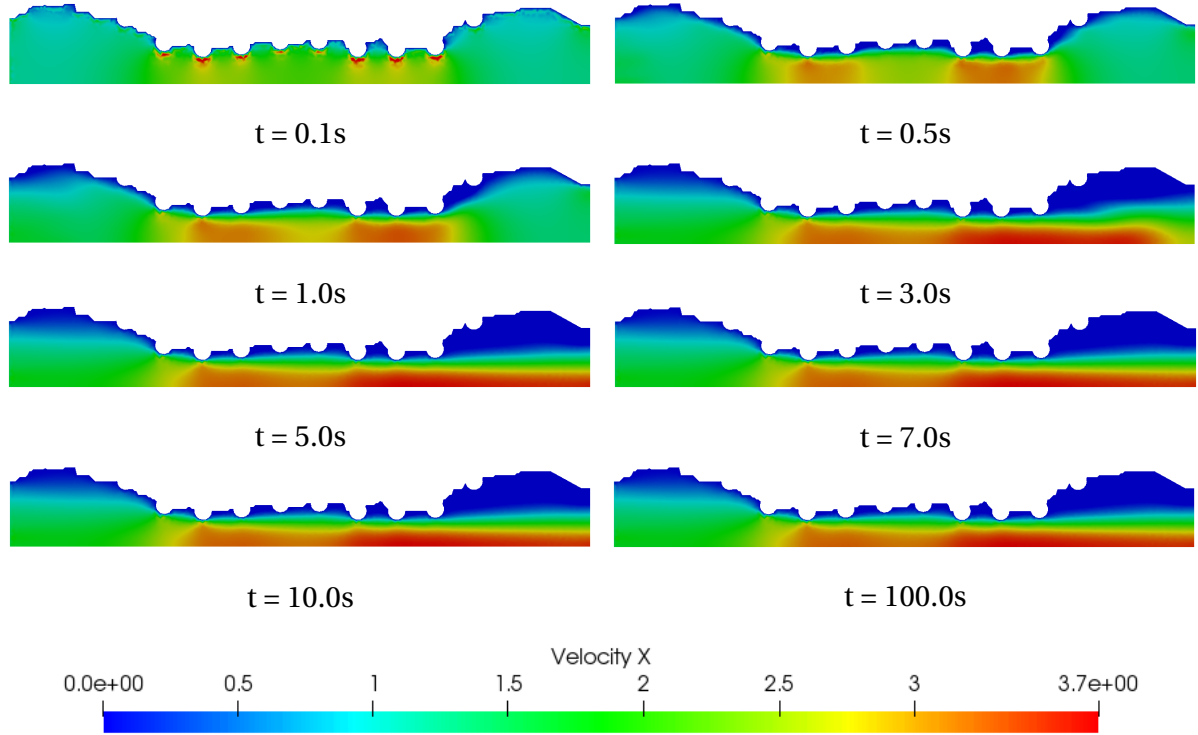

Figure 38: Temporal and spatial evolution of the velocity field for real channel with drug-eluting stent.

The Figure 39 to Figure 42 show the temporal and spatial evolution of the concentration field for several *Schmidt* number, such as: 1, 10, 100 and 1000 respectively. The concentration field is represented with the non-dimensional values where the red color represents 100% and

the blue color represents 0% of the diffused concentration in the bloodstream. It is possible to observe that the concentration field dynamic is similar than the curved channel, no significant differences from the previous one for each Schmidt number. Only in some regions that the concentration field become more diffuse as a consequence of the velocity field decrease due to irregular geometry close to the semi-circles of the stents.
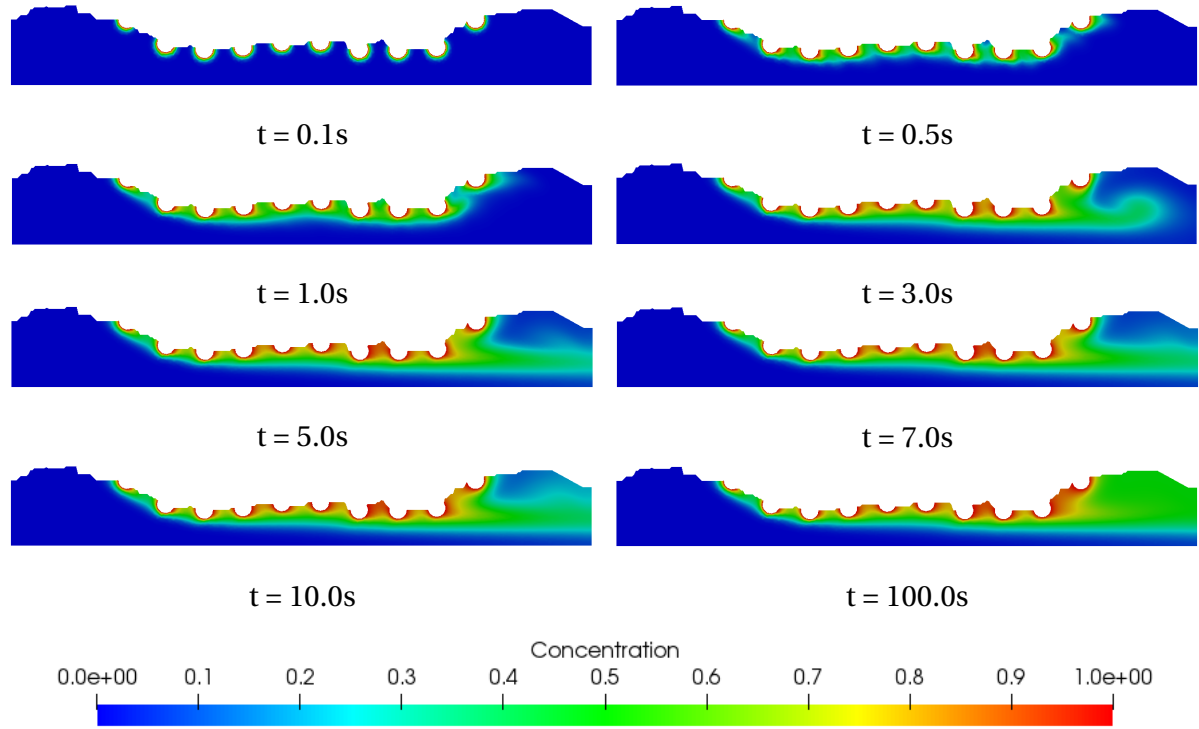

Figure 39: Temporal and spatial evolution of the concentration fiel for real channel with drug-eluting stent and $Sc = 1$.
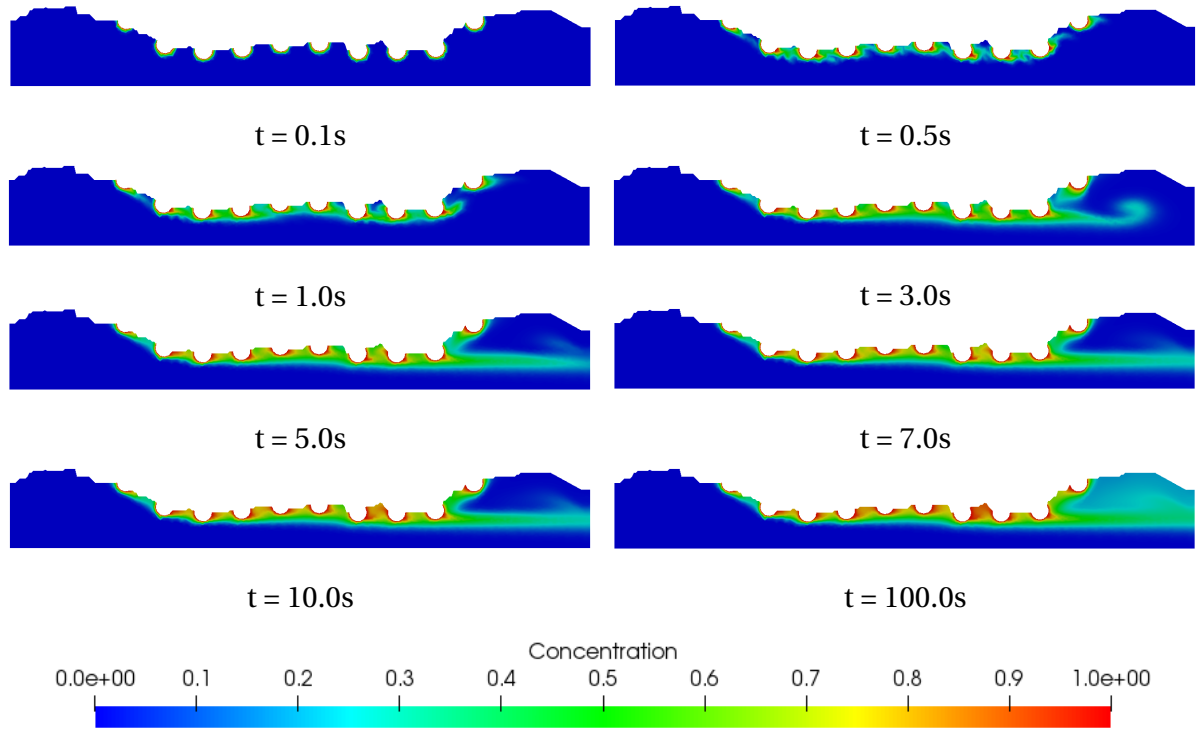
Figure 40: Temporal and spatial evolution of the concentration fiel for real channel with drug-eluting stent and $Sc = 10$.
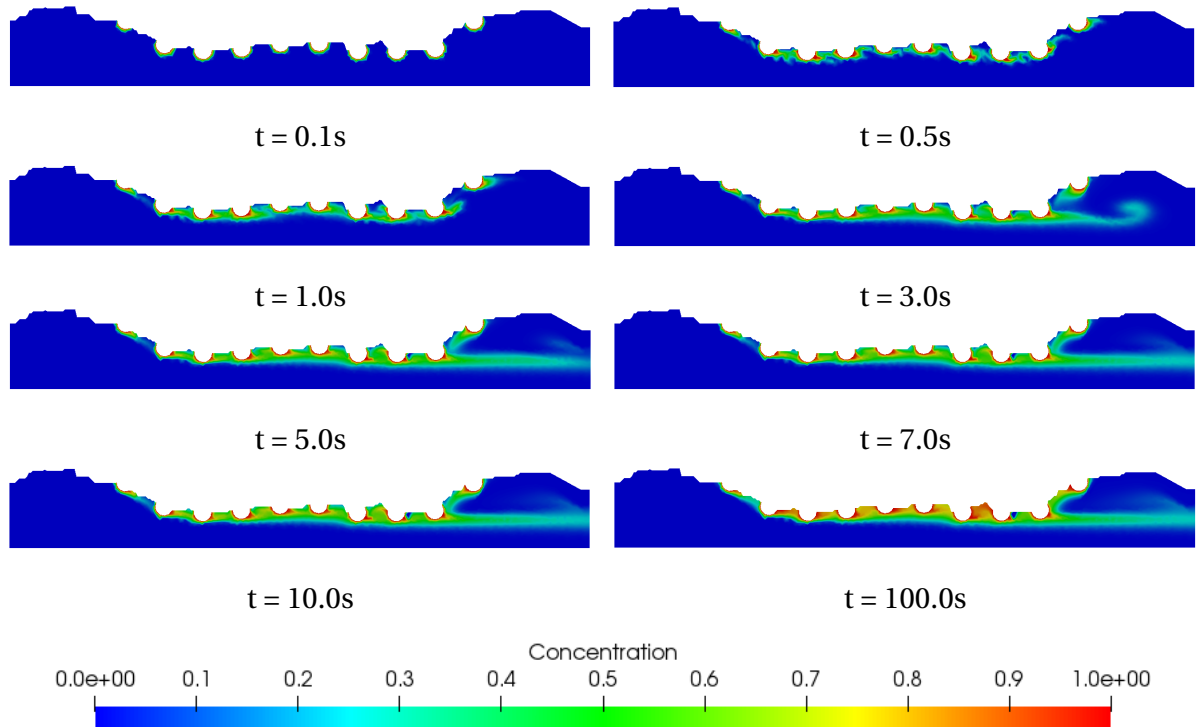

Figure 41: Temporal and spatial evolution of the concentration fiel for real channel with drug-eluting stent and $Sc = 100$.
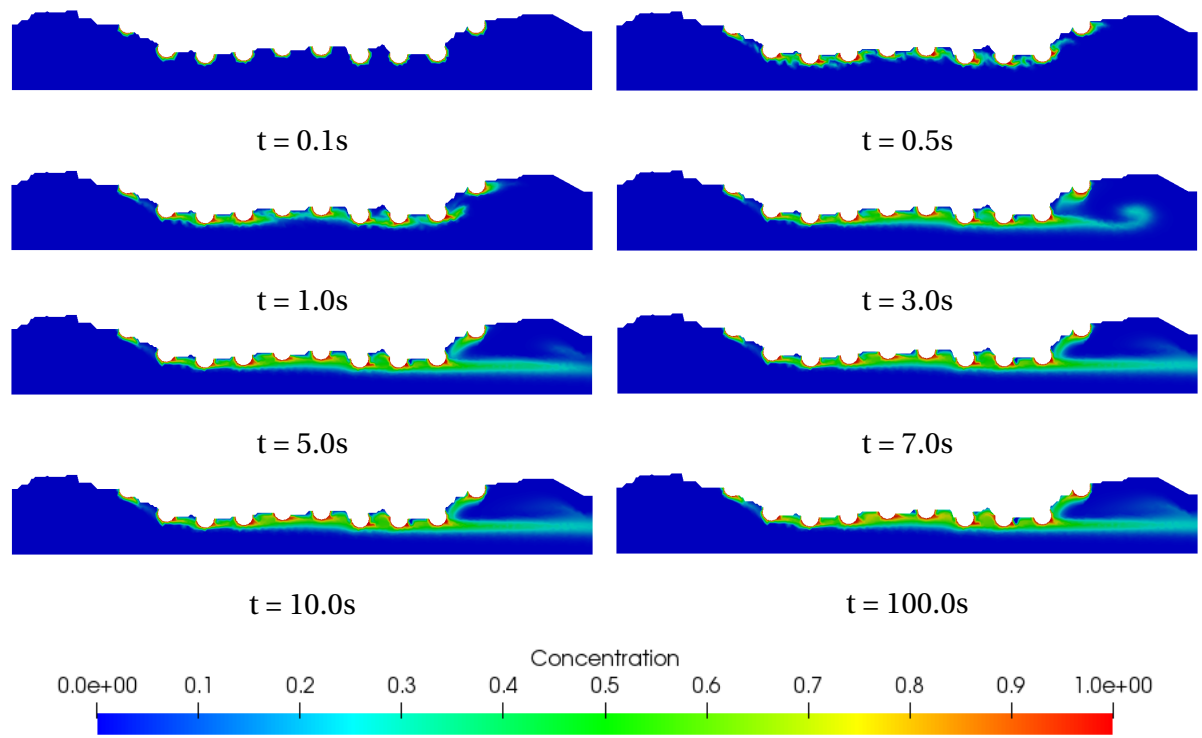
Figure 42: Temporal and spatial evolution of the concentration fiel for real channel with drug-eluting stent and $Sc = 1000$.

CONCLUSION

In this work, the Navier-Stokes equation according to the vorticity-streamfunction formulation with the species transport equation was presented in a Finite Element Method approach using the Arbitrary Lagrangian-Eulerian description. The governing equations were discretized in spatial domain by Galerkin Method and in time domain by semi-Lagrangian Method. As the vorticity-streamfuntion formulation does not have the coupling between the velocity and pressure fields, we can use the linear triangular element. In addition, the unknowns are scalar, in contrast to the primitive variables that are vectorial fields. In this way, a smooth implementation the numerical code is possible.

A computational code was developed in a high-level programming language using the object orientation paradigm (OOP), providing a platform for numerical simulations of the drug transport in bloodstream. The simulator is also able to describe in detail problems involving the flow of Newtonian fluids with scalar transport (such that the temperature) due to the generalized construction of the code. The average computational cost of each simulation process was measured for further improvement.

The numerical code showed satisfactory results compared to the analytical solutions of *Couette Flow*, *Poiseuille Flow* and *Half Poiseuille Flow* where the free-slip condition on the symmetry axis was applied. Moreover, the mesh convergence test was performed where it was concluded that the numerical code error decreases linearly as the mesh is refined. The *lid-driven cavity flow* was also simulated where the results were compared with those presented by Ghia et al. (1982) [?] and Marchi et al. (2009) [?] for several Reynolds number. Finally, the comparison between the semi-Lagrangian Method in an unstructured linear and a quadratic triangular mesh was presented for a purely advection flow of a parabolic function where it was possible to observe the effectiveness of the quadratic triangular element compared to linear element for decrease the numerical diffusion in high Reynolds number. Thus, it was possible to validate the numerical code for two-dimensional simulations in cartesian coordinates, using the Vorticity-Streamfunction Formulation with the Species Transport Equation in an ALE-FE context and sumitted to the Dirichlet boundary condition.

The other goal of this work was to understand the dynamics of blood flow in a coronary artery with a drug-eluting stent placed for several Schmidt number. Thus, the simulation for two geometries modeled in a two-dimensional cartesian coordinates was presented. The

profile of the velocity field was shown for the two proposed geometries where it was possible to observe the horizontal velocity increase close to symmetric axis. In addition, the curved channel model shown an acceptable approach because the maximum horizontal velocity had a difference less than 1% from the real channel. The simulation was done using several *Schmidt* numbers, such as 1, 10, 100 and 1000. It was possible to verify in the simulation that the number of *Schmidt* directly influences the transport of the drug in the bloodstream. For high values of the *Schmidt* number, the transport of chemical species becomes purely convective and the diffusion in blood flow is decreased.

The following futher developments are proposed:

- Increase assembly performance

- Use of primitive variables in the Navier-Stokes equation in a 3D approach

- Blood flow model as a multiphase problem

- Blood model as a non-Newtonian fluid