



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

Leandro Marques dos Santos

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

Rio de Janeiro, Brazil

2020

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 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

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

S237 Santos, Leandro Marques dos
Simulação Numérica em Elementos Finitos do Escoamento em
Artéria Coronária / Leandro Marques dos Santos – 2018.
138 f.

Orientador: Gustavo Rabello dos Anjos
Trabalho de Conclusão de Curso apresentado à Universidade do
Estado do Rio de Janeiro, Faculdade de Engenharia, para a obten-
ção do grau de bacharel em Engenharia Mecânica.

1. Formulação Corrente-Vorticidade - Monografias. 2. Método
dos Elementos Finitos - Monografias. 3. Esquema Taylor-Galerkin -
Monografias. 4. Stent Farmacológico - Monografias. 5. Ateroscle-
rose - Monografias. I. Anjos, Gustavo Rabello dos. II. Universidade
do Estado do Rio de Janeiro. Faculdade de Engenharia. III. Título.

CDU 621

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

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

This work was developed at the Numerical Simulations Laboratory (LEN) of the Environmental Simulations in Reservoirs and Study Group (GESAR) of State University of Rio de Janeiro (UERJ).

Initially, I would like to register my gratitude and great admiration for my advisor, Prof. Gustavo Rabello dos Anjos, for his teachings, instructions and attention in preparing this work. Also, I would like to thank my co-Advisor Prof. José da Rocha Miranda Pontes for support and development in my academic background. To the Gesar Research Group, in particular to Prof. Norberto Mangiavacchi for providing all infrastructure necessary for the elaboration of this work.

My thanks to my colleagues of LEN, Livia Correa, Luis Carnevale, Haroldo Rufino, Gabriel Oliveira and Daniel Coelho, to contributing in several ways to complete this objective and made my days more pleasant.

To my family for all encouragement and support. To my parents, Ana Marques and Edson Santos, for careful. To my brother, Pedro Marques, for friendship. To my wife, Letícia Marques, who this work is dedicated, for all attention, encouragement and friendship that has deposited all days.

To particular, to the Creator, who provides all inspiration source and existence.

ABSTRACT

Marques, Leandro *An ALE-FE Method for Vorticity-Streamfunction Formulation with Species Transport Equation.* 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 using Vorticity-Streamfunction Formulation in an Arbitrary Lagrangian-Eulerian (ALE) approach. The blood was modeled as single-phase, incompressible and newtonian fluid and. The Navier-Stokes equation is shown according to the vorticity-streamfunction formulation with pecies transport equation. The Finite Element Method (FEM) is used to solve the governing equations where the Galerkin formulation was used to discretized the equations in space and the semi-Lagrangian method was used to discretized the material derivative using first order backward difference scheme. The linear systems was solved using Conjugate Gradient Iterative Method.

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 Formulação Corrente-Vorticidade com a Equação de Transporte de Espécie Química.* xx. 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 usando a Formulação Corrente-Vorticidade em uma abordagem Lagrangeana-Euleriana Arbitrária (ALE). O sangue 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*. Os sistemas lineares foram resolvidos utilizando o Método Iterativo Gradientes Conjugados.

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

LIST OF FIGURES

Figure 1	Comparison PTCA procedure: (a) balloon and (b) stent.....	14
Figure 2	Several models of drug-eluting stent [1].	16
Figure 3	Variação da vorticidade em um passo de tempo	52
Figure 4	Linear Triangular Element.....	56
Figure 5	Quad Triangular Element	57
Figure 6	Mini Triangular Element	57
Figure 7	Simplified Class Diagram	61
Figure 8	Edges of a node.....	69
Figure 9	Solve algorithm for Vorticity-Streamfunction Formulation with Species Transport Equation.....	70
Figure 10	Couette Flow.....	76
Figure 11	Unsteady velocity profile when $Re = 100$ and the comparison between the numerical and analytical solution for Couette flow.	77
Figure 12	Poiseuille Flow	77
Figure 13	Unsteady velocity profile when $Re = 100$ and the comparison between the numerical and analytical solution for Poiseuille flow.	78
Figure 14	Convergence order in log-log scale: It is estimated the the relative error of numerical solution has first order convergence.	80
Figure 15	Half Poiseuille flow	80
Figure 16	Unsteady velocity profile when $Re = 100$ and the comparison between the numerical and analytical solution for Half Poiseuille flow.	81
Figure 17	Lid-driven Cavity Flow	82
Figure 18	The tangential velocity u profile in central line of cavity ($x = 0.5$) for several Reynolds number: (a) 10 (b) 100 (c) 400 (d) 1000.	83
Figure 19	The normal velocity v profile in central line of cavity ($y = 0.5$) for several Reynolds number: (a) 10 (b) 100 (c) 400 (d) 1000.	84
Figure 20	Transport of a scalar in Pure Advection Flow.	85

Figure 21	Comparison of c profile for the Galerkin e Taylor-Galerkinmethod in several positions on the axis of rotation: (a) initial, (b) 1/4 rotation, (c) 1/2 rotation and (d) 3/4 rotation.	86
Figure 22	Spurious oscillations for the Galerkin method in several positions of the axis of rotation: (a) initial, (b) 1/4 rotation, (c) 1/2 rotation and (d) 3/4 rotation. ...	87
Figure 23	Spurious oscillations for the Taylor-Galerkin method in several positions of the axis of rotation: (a) initial, (b) 1/4 rotation, (c) 1/2 rotation and (d) 3/4 rotation.	88
Figure 24	Non-dimensional Domaian for blood flow in coronary artery The radius used was $R = 1$ and and the lumen length was $L = 10R$. (a) Curved Channel (b) Curved Channel with Drug-Eluting Stent (c) Real Channel (d) Real Channel with Drug-Eluting Stent.	89
Figure 25	The unsteady velocity profile for the curved channel.	90
Figure 26	Time and space evolution of the velocity field for curved channel.....	91
Figure 27	The unsteady velocity profile for curved channel with drug-eluting stent.....	92
Figure 28	Time and space evolution of the velocity field for curved channel with drug-eluting stent.	93
Figure 29	Time and space evolution of the concentration fiel for curved channel with drug-eluting stent with $Sc = 1$	94
Figure 30	Time and space evolution of the concentration fiel for curved channel with drug-eluting stent with $Sc = 10$	94
Figure 31	The unsteady velocity profile for the real channel.....	95
Figure 32	Time and space evolution of the velocity field for real channel.	96
Figure 33	The unsteady velocity profile for real channel with drug-eluting stent.....	97
Figure 34	Time and space evolution of the velocity field for real channel with drug-eluting stent.	98
Figure 35	Time and space evolution of the concentration field for real channel with drug-eluting stent when the $Sc = 1$	99
Figure 36	ime and space evolution of the concentration field for real channel with drug-eluting stent when the $Sc = 10$	99

LIST OF TABLES

Table 1	Average processing time for mesh import in several unstructured linear triangular meshes.....	62
Table 2	Average processing time for global matrices assembly in several unstructured linear triangular meshes	64
Table 3	Average processing time for Dirichlet condition in several unstructured linear triangular meshes.....	67
Table 4	The relative error of numerical solution for several elements numbers in an unstructured linear triangular mesh.	79

CONTENTS

INTRODUCTION	11
1 LITERATURE REVIEW	13
1.1 Introduction.....	13
1.2 Drug-Eluting Stent	13
1.3 Finite Element Method - Convection-Diffusion Equation.....	17
2 GOVERNING EQUATIONS	20
2.1 Introduction.....	20
2.2 Arbitrary Lagrangian-Eulerian.....	20
2.3 Mass Conservation	22
2.4 Linear Momentum Conservation.....	25
2.5 Species Transport Conservation	29
2.6 Non-dimensionalization	31
2.7 Vorticity-Streamfunction Formulation	33
2.8 Generic Initial and Boundary Conditions.....	36
3 FINITE ELEMENT METHOD.....	38
3.1 Introduction.....	38
3.2 Strong Formulation	38
3.3 Weak Formulation.....	39
3.4 Galerkin Method.....	45
3.5 Semi-Lagrangian Method	51
3.6 Mesh Elements	56
4 COMPUTATIONAL CODE	61
4.1 Introduction.....	61
4.2 Mesh Import.....	62
4.3 Matrix Assembly	63
4.4 Boundary Condition Apply	64
4.5 Solve Algorithm.....	70

5	VALIDATIONS	75
5.1	Introduction	75
5.2	Couette Flow.....	76
5.3	Poiseuille Flow	77
5.4	Half Poiseuille Flow	80
5.5	Lid-Driven Cavity.....	82
5.6	Pure Advection Flow	84
6	RESULTS	89
6.1	Introduction	89
6.2	Curved Channel.....	90
6.3	Curved Channel with Drug-Eluting Stent	92
6.4	Real Channel	94
6.5	Real Channel with Drug-Eluting Stent	97
	CONCLUSION	100
	BIBLIOGRAPHY	102
	APÊNDICE - PUBLICAÇÕES	108

INTRODUCTION

According to the World Health Organization (WHO), more people die each year from cardiovascular disease (CVD) than any other cause in the world every year [2]. It is estimated that 17.7 million people died from CVD in 2015, representing 31% of all deaths in the world. About 40% 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 [3]. 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 [?]. The main objectives of this work plan are the development of a Finite Element code using the Lagragian-Eulerian Arbitrary (ALE) description for the linear momentum conservation and for species transport equationin an incompressible, one-phase and newtonian fluid, in addtion to know the dynamics of blood flow in a coronary artery with atherosclerosis and drug-eluting stent placed.

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 diffusive coefficient was approximated as constante. The Navier-Stokes equation is shown according to the vorticity-streamfunction formulation with species transport equation without internal source term.

The domain were discretized over an unstructured triangular mesh generated by GMSH open source [4] and the Finite Element Method (FEM) were used. 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 [5] [6]. The Navier-Stokes and Species Transport equations were discretized in time using the Taylor series and the semi-Lagragian Methodo [?] was used in order to reduce spurious oscillations that are usually seen in the diffusion-convection equations. Finally, Galerkin Method was used to discretize the equations in space.

The computational development was done in Python [7] using Object-Oriented Paradigm (OOP) and the Chapter 4 presents the simulation process scripts in addition to the solution algorithm. 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*. The horizontal and vertical velocities in *Lid-Driven Cavity* was compared with those presented by *Ghia et al.* [8] and *Marchi et al.* [9]. Then, the comparison of the Taylor-Galerkin and semi-Lagrangian Methods is presented when spurious oscillations are present in a pure convective flow.

The blood flow hydrodynamics and the species chemical transport in coronary artery were simulated in 4 geometries types as suggested by *Wang et al.* [10], but with some modifications to the cartesian coordinates. These geometries types consist of one: (1) coronary artery with atherosclerosis with 40% lumen obstruction; (2) coronary artery with atherosclerosis and with drug-eluting stent placed; (3) real coronary artery with atherosclerosis with 40% lumen obstruction; (4) 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)* [11].

This work was organized as follows:

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

1 LITERATURE REVIEW

1.1 Introduction

In this chapter, the literature review is presented and discussed, addressing 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) [12], cardiovascular diseases have remained the leading deaths causes globally in the last 15 years.

In 1964, Dotter and Judkins [13] introduced a new technique for obstructed femoral artery treatment due to atherosclerosis. This technique is known as *percutaneous transluminal angioplasty* and it consists of a simple and minimally invasive procedure, allowing execution by any doctor familiar with a vascular catheterization. Such procedure is presented applicable to other arteries, including the coronary artery.

In 1979, Gruntzig, Senning and Seigenthaler [14] 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. [15] 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. [16] 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 ?? presents the comparison between PTCA procedures using the balloon and stent.

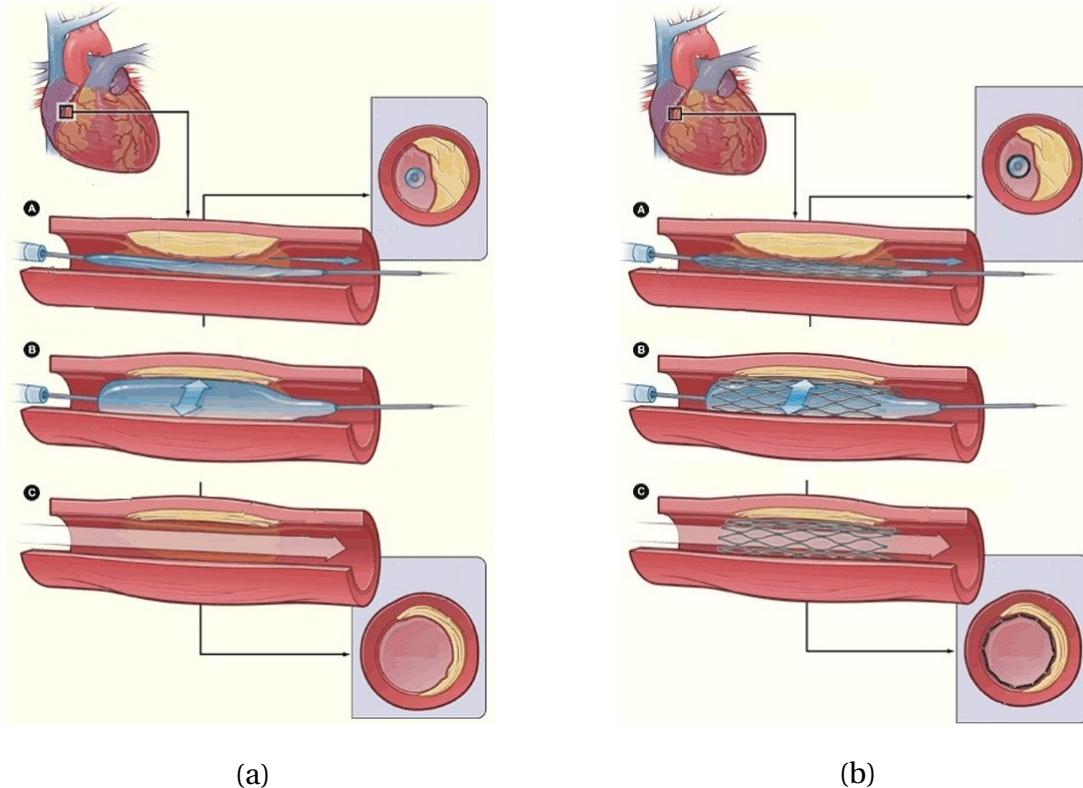


Figure 1: Comparison PTCA procedure: (a) balloon and (b) stent.

In 2001, Hwang, Wu and Edelman [17] 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. [18] presented a complete overview of mathematical models and finite element numerical simulation applied to the modelling of drug eluting stents 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 a small fraction of the total amount drug stored into the stent was effectively delivered to the artery.

In 2014, Bozsak, Chomaz and Barakat [19] 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. [20] present a fluid-structure interaction between a curved coronary artery with an implanted stent, pulsatile blood flow and heart contractions. A 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 these 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. [10] 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. [21] 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 governing 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 diffusion coefficient 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 [22] 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* [23], *Taxus I and Taxus II* [24] [25], *C-Sirius* [26], *Smart* [27] 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.

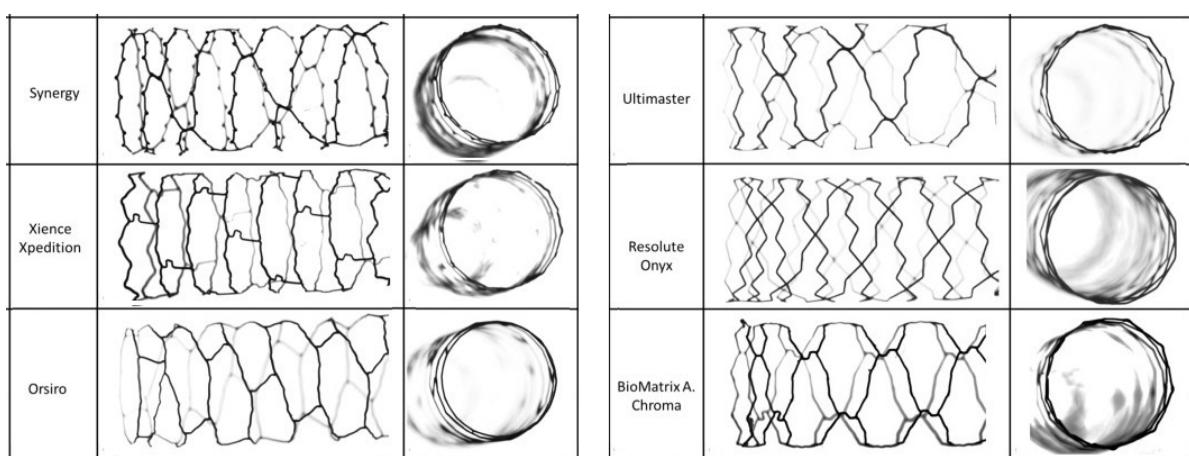


Figure 2: Several models of drug-eluting stent [1].

1.3 Finite Element Method - Convection-Diffusion Equation

The mathematical basis for the Finite Element Method begins in 1909 with Ritz [28] 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) [29] 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) [30] applied variational formulation to a domain discretized by triangular elements. In 1965, Zienkiewicz and Cheung [31] 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 [32] where it is shown that the *upwind* effect helped to reduce these oscillations.

In 1976, Christie et al. [33] 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 [34] 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 [35] proposed a new formulation that consists of modifying the weight functions so that the diffusion operator acts only in the flow direction. This procedure appears in order 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 [36] 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*.

In 1984, Donea [37] 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 precision and low numerical diffusion. Although the Taylor-Galerkin method is conditionally stable, the computational implementation is easy.

In the same year, Lohner et al. [38] 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) [39]. In this paper, the authors present an analysis of several methods for solving unsteady problems governed by advection equations. In comparison to the Petrov-Galerkin and SUPG schemes, 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 are of a much simpler implementation and 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) [40]. For current work, the semi-Lagrangian method was chosen for decrease spurious oscillations and it will be described in the following sections.

2 GOVERNING EQUATIONS

2.1 Introduction

In this work, the fluid is considered as a continuum medium. 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. 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 for a fluid element. 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

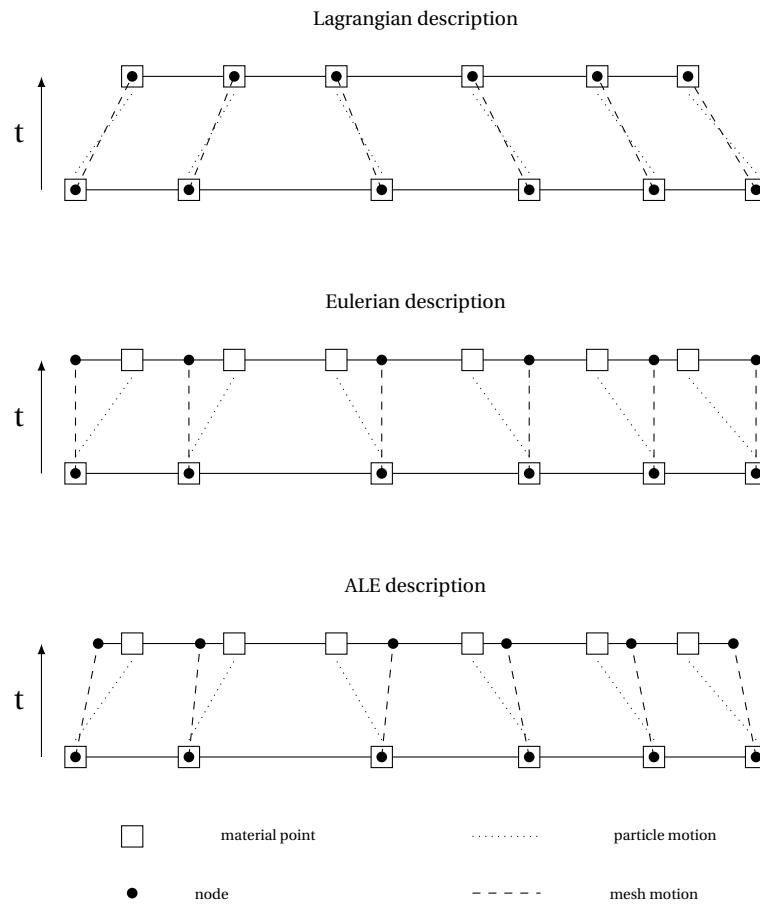
The choice of the kinematical description of the continuum is extremely important for the development of a computational code, since it directly affects the accuracy of the numerical result. In the literature, two classic descriptions are commonly used, namely Lagrangian and Eulerian.

The Lagrangian description is one where each computational mesh node moves at the same velocity as the material point, as can be seen in Fig. (3.1a). 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 insertion 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 Fig. (3.1b). 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. It is 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 Fig. (3.1c). 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.



The ALE description was first implemented in the finite difference method, as observed in the Hirt et al. (1974) [41] and was subsequently adopted in the finite elements context, as presented by Donea (1982) [42]. 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 figure 1. However, it is possible to correlate these frameworks. For instance, if the operator Z 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 operator Y 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) [43] and Hughes (1981) [44].

2.3 Mass Conservation

As presented by Batchelor (1967) [45], the principle of mass conservation without source term establishes that:

$$\left[\begin{array}{c} \text{Mass accumulation rate} \\ \text{on control volume} \end{array} \right] = - \left[\begin{array}{c} \text{Mass flux crossing} \\ \text{the boundary} \end{array} \right]$$

Mathematically, the mass accumulation rate within on volume can be represented by:

$$\int_V \frac{\partial}{\partial t} dm \quad (2.1)$$

where the infinitesimal mass dm is defined as $dm = \rho dV$. Replacing it in Eq. 2.1 and considering that the control volume does not vary over time, we have:

$$\int_V \frac{\partial}{\partial t} dm = \int_V \frac{\partial}{\partial t} (\rho dV) = \int_V \frac{\partial \rho}{\partial t} dV + \int_V \rho \frac{\partial dV}{\partial t} = \int_V \frac{\partial \rho}{\partial t} dV \quad (2.2)$$

The mass flux crossing boundary can be mathematically represented by:

$$\oint_S \rho \mathbf{c} \cdot \mathbf{n} dA \quad (2.3)$$

Thereby, according to mass conservation theorem:

$$\int_V \frac{\partial \rho}{\partial t} dV = - \oint_S \rho \mathbf{c} \cdot \mathbf{n} dA \quad (2.4)$$

Applying the *Gauss theorem* on surface integral:

$$\int_V \frac{\partial \rho}{\partial t} dV = - \int_V \nabla \cdot (\rho \mathbf{c}) dV \quad (2.5)$$

that is:

$$\int_V \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{c}) \right] dV = 0 \quad (2.6)$$

Whereas the $dV \neq 0$, the Eq. 2.6 can be presented as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{c}) = 0 \quad (2.7)$$

where ρ is density, \mathbf{c} is relative velocity and it is relationship between the material velocity \mathbf{v} and the computational mesh velocity $\hat{\mathbf{v}}$, that is, $\mathbf{c} = \mathbf{v} - \hat{\mathbf{v}}$. In addition, ∇ is Del operator whose components are $\nabla = [\partial/\partial x, \partial/\partial y]$, x and y are coordinates components and t is time. The Eq. 2.7 is known as *Continuity Equation* for Arbitrary Lagrangian-Eulerian description [42]. Developing the equation, we have:

$$\frac{\partial \rho}{\partial t} + \mathbf{c} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{v} = 0 \quad (2.8)$$

According to fluid incompressible assumption, the density does not depend 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:

$$\rho \nabla \cdot \mathbf{v} = 0 \quad (2.9)$$

that is:

$$\nabla \cdot \mathbf{v} = 0 \quad (2.10)$$

This equation is the *continuity equation* for an incompressible flow.

2.4 Linear Momentum Conservation

The same concept of mass conservation is applied to the linear momentum conservation. Therefore, the principle of linear momentum conservation establishes that:

$$\begin{bmatrix} \text{Linear momentum} \\ \text{accumulation rate} \\ \text{on control volume} \end{bmatrix} = - \begin{bmatrix} \text{Linear momentum} \\ \text{flux crossing} \\ \text{the boundary} \end{bmatrix} + \begin{bmatrix} \text{The resultant of} \\ \text{body force and} \\ \text{surface force} \end{bmatrix}$$

Mathematically, linear momentum accumulation rate on control volume can be presented to represented by:

$$\int_V \frac{\partial}{\partial t} (\rho \mathbf{v}) dV \quad (2.11)$$

The linear momentum flux crossing the boundary can be represented mathematically by:

$$\oint_S \rho \mathbf{v} \cdot \mathbf{n} dA \quad (2.12)$$

The resultant force is consists of surface force and body force. The surface force can be represented by:

$$\oint_S \sigma \cdot \mathbf{n} dA \quad (2.13)$$

where σ is the stress tensor. Whereas, the body is represend by:

$$\int_V \rho \mathbf{g} dV \quad (2.14)$$

where \mathbf{g} is gravity vector. Thereby, according to principle of linear momentum conservation:

$$\int_V \frac{\partial}{\partial t}(\rho\mathbf{v})dV = -\oint_S \rho\mathbf{vc} \cdot \mathbf{n}dA + \oint_S \sigma \cdot \mathbf{n}dA + \int_V \rho\mathbf{gdV} \quad (2.15)$$

Applying the *Gauss Theorem* on surface integrals:

$$\int_V \frac{\partial}{\partial t}(\rho\mathbf{v})dV = -\int_V \nabla \cdot (\rho\mathbf{vc})dV + \int_V \nabla \cdot \sigma dV + \int_V \rho\mathbf{gdV} \quad (2.16)$$

that is:

$$\int_V \left[\frac{\partial}{\partial t}(\rho\mathbf{v}) + \nabla \cdot (\rho\mathbf{vc}) - \nabla \cdot \sigma - \rho\mathbf{g} \right] dV = 0 \quad (2.17)$$

Whereas the $dV \neq 0$, the Eq. 2.17 can be presented as:

$$\frac{\partial}{\partial t}(\rho\mathbf{v}) + \nabla \cdot (\rho\mathbf{vc}) - \nabla \cdot \sigma - \rho\mathbf{g} = 0 \quad (2.18)$$

that is:

$$\frac{\partial}{\partial t}(\rho\mathbf{v}) + \nabla \cdot (\rho\mathbf{vc}) = \nabla \cdot \sigma + \rho\mathbf{g} \quad (2.19)$$

Developing the left hand side of equation, we have:

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \frac{\partial \rho}{\partial t} + \rho \mathbf{c} \cdot \nabla \mathbf{v} + \mathbf{v} \nabla \cdot (\rho \mathbf{c}) = \rho \left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right] + \mathbf{v} \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{c}) \right] \quad (2.20)$$

The last term of above equation is null because the *continuity equation* (Eq. 2.7). Thus, the linear momentum equation can be rewritten as:

$$\rho \left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{v} \right] = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} \quad (2.21)$$

The stress tensor $\boldsymbol{\sigma}$ can be split into two tensors:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau} \quad (2.22)$$

where, p is pressure field, \mathbf{I} is the identity matrix and $\boldsymbol{\tau}$ is deviatoric stress. Replacing them in Eq. 2.21, we have:

$$\rho \left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{v} \right] = \nabla \cdot [-p\mathbf{I} + \boldsymbol{\tau}] + \rho \mathbf{g} \quad (2.23)$$

that is:

$$\rho \left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{v} \right] = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{g} \quad (2.24)$$

The deviatoric stress $\boldsymbol{\tau}$ 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:

$$\boldsymbol{\tau} = \mu [\nabla \mathbf{v} + (\nabla \mathbf{v})^T] + \lambda \mathbf{I} \nabla \cdot \mathbf{v} \quad (2.25)$$

where μ is dynamic viscosity of fluid, λ is known as the second viscosity coefficient and \mathbf{I} is

identity matrix. Replacing them in Eq. 2.24, we have:

$$\rho \left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{v} \right] = -\nabla p + \nabla \cdot [\mu [\nabla \mathbf{v} + (\nabla \mathbf{v})^T] + \lambda \mathbf{I} \nabla \cdot \mathbf{v}] + \rho \mathbf{g} \quad (2.26)$$

that is:

$$\rho \left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{v} \right] = -\nabla p + \nabla \cdot [\mu [\nabla \mathbf{v} + (\nabla \mathbf{v})^T]] + \nabla \cdot [\lambda \mathbf{I} \nabla \cdot \mathbf{v}] + \rho \mathbf{g} \quad (2.27)$$

Whereas the dynamic viscosity μ does not depends on coordinates, we have:

$$\rho \left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{v} \right] = -\nabla p + \mu [\nabla \cdot \nabla \mathbf{v} + \nabla \cdot (\nabla \mathbf{v})^T] + \nabla \cdot [\lambda \mathbf{I} \nabla \cdot \mathbf{v}] + \rho \mathbf{g} \quad (2.28)$$

that is:

$$\rho \left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{v} \right] = -\nabla p + \mu [\nabla^2 \mathbf{v} + \nabla (\nabla \cdot \mathbf{v})] + \nabla \cdot [\lambda \mathbf{I} \nabla \cdot \mathbf{v}] + \rho \mathbf{g} \quad (2.29)$$

According to Eq. 2.10, we have:

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{v} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{g} \quad (2.30)$$

where ν is the kinematic viscosity of fluid. The Eq. 2.30 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 [42].

2.5 Species Transport Conservation

The species transport conservation establish that:

$$\begin{bmatrix} \text{Chemical species} \\ \text{accumulation rate} \\ \text{on control volume} \end{bmatrix} = - \begin{bmatrix} \text{Chemical species} \\ \text{flux crossing} \\ \text{the boundary} \end{bmatrix} + \begin{bmatrix} \text{The resultant of} \\ \text{chemical species} \\ \text{source rate} \end{bmatrix}$$

Mathematically, chemical species accumulation on control volume can be represented by:

$$\int_V \frac{\partial e}{\partial t} dV \quad (2.31)$$

The chemical species flux crossing the boundary can be splitted into two flux:

$$\oint_S e\mathbf{c} \cdot \mathbf{n} dA - \oint_S D\nabla e \cdot \mathbf{n} dA \quad (2.32)$$

where D is coefficient of chemical species diffusion. The resultant of chemical species source rate on volume can be represented by:

$$\int_V \dot{R} dV \quad (2.33)$$

Thereby, according to species transport conservation:

$$\int_V \frac{\partial e}{\partial t} dV = - \oint_S e\mathbf{c} \cdot \mathbf{n} dA + \oint_S D\nabla e \cdot \mathbf{n} dA + \int_V \dot{R} dV \quad (2.34)$$

Applying the *Gauss Theorem* on surface integrals:

$$\int_V \frac{\partial e}{\partial t} dV = - \int_V \nabla \cdot (e\mathbf{c}) dV + \int_V \nabla \cdot (D\nabla e) dV + \int_V \dot{R} dV \quad (2.35)$$

that is:

$$\int_V \left[\frac{\partial e}{\partial t} + \nabla \cdot (e\mathbf{c}) - \nabla \cdot (D\nabla e) - \dot{R} \right] dV = 0 \quad (2.36)$$

Whereas the $dV \neq 0$, the Eq. 2.36 can be represented by:

$$\frac{\partial e}{\partial t} + \nabla \cdot (e\mathbf{c}) - \nabla \cdot (D\nabla e) - \dot{R} = 0 \quad (2.37)$$

that is:

$$\frac{\partial e}{\partial t} + \nabla \cdot (e\mathbf{c}) = \nabla \cdot (D\nabla e) + \dot{R} \quad (2.38)$$

Developing the left hand side of equation, we have:

$$\frac{\partial e}{\partial t} + \mathbf{v} \cdot \nabla e + e \nabla \cdot \mathbf{c} = \nabla \cdot (D\nabla e) + \dot{R} \quad (2.39)$$

The last term of above equation is null because the fluid incompressibility assumption (Eq. 2.10), thus:

$$\frac{\partial e}{\partial t} + \mathbf{c} \cdot \nabla e = \nabla \cdot (D\nabla e) + \dot{R} \quad (2.40)$$

Considering the diffusion coefficient is constant and without chemical species generation, the species transport equation can rewritten as:

$$\frac{\partial e}{\partial t} + \mathbf{c} \cdot \nabla e = D \nabla^2 e \quad (2.41)$$

The Eq. 2.41 is known as *Species Transport Equation* for an incompressible fluid, with constant diffusion coefficient and without chemical species generation for the Arbitrary Lagrangian-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:

$$\begin{aligned} p &= \rho_0 U^2 p^* & e &= (e_s - e_0) e^* + e_0 & v &= v_0 v^* & D &= D_0 D^* & x &= L x^* \\ \mathbf{v} &= U \mathbf{v}^* & \mathbf{g} &= g_0 \mathbf{g}^* & \rho &= \rho_0 \rho^* & \nabla &= \frac{1}{L} \nabla^* & t &= \frac{L}{U} t^* \end{aligned}$$

where the asterisk identify the non-dimensional unknowns. Replacing the above parameters in Eq. 2.10, we have:

$$\frac{U}{L} \nabla^* \cdot \mathbf{v}^* = 0 \quad (2.42)$$

Multiplying both sides by U/L :

$$\nabla^* \cdot \mathbf{v}^* = 0 \quad (2.43)$$

A similar procedure is performed in Eq. 2.30, that is:

$$\frac{U^2}{L} \frac{\partial \mathbf{v}^*}{\partial t^*} + \frac{U^2}{L} \mathbf{c}^* \cdot \nabla^* \mathbf{v}^* = -\frac{U^2}{L} \frac{1}{\rho^*} \nabla^* p^* + \frac{\nu_0 U}{L^2} \nu^* \nabla^{*2} \mathbf{v}^* + g_0 \mathbf{g}^* \quad (2.44)$$

Multiplying both sides by L/U^2 :

$$\frac{\partial \mathbf{v}^*}{\partial t^*} + \mathbf{c}^* \cdot \nabla^* \mathbf{v}^* = -\frac{1}{\rho^*} \nabla^* p^* + \frac{\nu_0}{UL} \nu^* \nabla^{*2} \mathbf{v}^* + \frac{g_0 L}{U^2} \mathbf{g}^* \quad (2.45)$$

that is:

$$\frac{\partial \mathbf{v}^*}{\partial t^*} + \mathbf{c}^* \cdot \nabla^* \mathbf{v}^* = -\nabla^* p^* + \frac{\nu_0}{UL} \nabla^{*2} \mathbf{v}^* + \frac{g_0 L}{U^2} \mathbf{g}^* \quad (2.46)$$

In the Eq. 2.41, a similar procedure is performed:

$$(e_s - e_0) \frac{U}{L} \frac{\partial e^*}{\partial t^*} + (e_s - e_0) \frac{U}{L} \mathbf{c}^* \cdot \nabla^* e^* = (e_s - e_0) \frac{D_0}{L^2} D^* \nabla^{*2} e^* \quad (2.47)$$

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

$$\frac{\partial e^*}{\partial t^*} + \mathbf{c}^* \cdot \nabla^* e^* = \frac{D_0}{UL} D^* \nabla^{*2} e^* \quad (2.48)$$

that is

$$\frac{\partial e^*}{\partial t^*} + \mathbf{c}^* \cdot \nabla^* e^* = \frac{D_0}{UL} \nabla^{*2} e^* \quad (2.49)$$

where, important non-dimensional groups are found in Eqs. 2.43, 2.46 and 2.49, 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. 2.43, 2.46 and 2.49 and removing the asterisk, we have:

$$\nabla \cdot \mathbf{v} = 0 \quad (2.50)$$

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{v} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{v} + \frac{1}{Fr^2} \mathbf{g} \quad (2.51)$$

$$\frac{\partial e}{\partial t} + \mathbf{c} \cdot \nabla e = \frac{1}{ReSc} \nabla^2 e \quad (2.52)$$

The Eqs. 2.50, 2.51 e 2.52 are, respectively, non-dimensional form of Continuity, Navier-Stokes and Species Transport equations for a newtonian and incompressible flow in an Arbitrary 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 2.51 the following vector identity:

$$\mathbf{c} \cdot \nabla \mathbf{v} = \nabla (\mathbf{c} \cdot \mathbf{v}) - \mathbf{c} \times \nabla \times \mathbf{v} \quad (2.53)$$

Therefore:

$$\frac{\partial \mathbf{v}}{\partial t} + \nabla (\mathbf{c} \cdot \mathbf{v}) - \mathbf{c} \times \nabla \times \mathbf{v} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{v} + \frac{1}{Fr^2} \mathbf{g} \quad (2.54)$$

Computing the curl on both sides of the above equation:

$$\nabla \times \frac{\partial \mathbf{v}}{\partial t} + \nabla \times \nabla(\mathbf{c} \cdot \mathbf{v}) - \nabla \times \mathbf{c} \times \nabla \times \mathbf{v} = -\nabla \times \nabla p + \frac{1}{Re} \nabla \times \nabla^2 \mathbf{v} + \frac{1}{Fr^2} \nabla \times \mathbf{g} \quad (2.55)$$

that is:

$$\frac{\partial}{\partial t} [\nabla \times \mathbf{v}] + \nabla \times \nabla(\mathbf{c} \cdot \mathbf{v}) - \nabla \times [\mathbf{c} \times \nabla \times \mathbf{v}] = -\nabla \times \nabla p + \frac{1}{Re} \nabla^2 [\nabla \times \mathbf{v}] + \frac{1}{Fr^2} \nabla \times \mathbf{g} \quad (2.56)$$

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 \mathbf{g} , are equal to zero. Thus, we have:

$$\frac{\partial}{\partial t} [\nabla \times \mathbf{v}] - \nabla \times [\mathbf{c} \times \nabla \times \mathbf{v}] = \frac{1}{Re} \nabla^2 [\nabla \times \mathbf{v}] \quad (2.57)$$

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

$$\frac{\partial w}{\partial t} - \nabla \times [\mathbf{c} \times w] = \frac{1}{Re} \nabla^2 w \quad (2.58)$$

The second term of left side in Eq. 2.58 can be replaced by following vectorial identity:

$$\nabla \times [\mathbf{c} \times w] = -\mathbf{c} \cdot \nabla w + w \cdot \nabla \mathbf{c} \quad (2.59)$$

Thus the Eq. 2.58 will be:

$$\frac{\partial w}{\partial t} + \mathbf{c} \cdot \nabla w - w \cdot \nabla \mathbf{c} = \frac{1}{Re} \nabla^2 w \quad (2.60)$$

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

to the velocity vector. Thus, the product $w \cdot \nabla \mathbf{c}$ will be canceled as presented by Pontes and Mangiavacchi (2016) [47]. Therefore:

$$\frac{\partial w}{\partial t} + \mathbf{c} \cdot \nabla w = \frac{1}{Re} \nabla^2 w \quad (2.61)$$

The Eq. 2.61 is known as *vorticity equation* for two-dimensional flows of a Newtonian and incompressible fluid in an Arbitrary Lagrangian-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. 2.10):

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (2.62)$$

The following relationship between the streamfunction and the velocity components can be defined so that Eq. 2.62 is satisfied:

$$u = \frac{\partial \psi}{\partial y} \quad v = -\frac{\partial \psi}{\partial x} \quad (2.63)$$

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

$$\nabla \times \mathbf{v} = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \quad (2.64)$$

Replacing the Eq. 2.63 in Eq. 2.64, we that:

$$w = -\frac{\partial}{\partial x} \frac{\partial v}{\partial x} - \frac{\partial}{\partial y} \frac{\partial u}{\partial y} \quad (2.65)$$

that is:

$$w = -\nabla^2 \psi \quad (2.66)$$

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

$$\frac{\partial w}{\partial t} + \mathbf{c} \cdot \nabla w = \frac{1}{Re} \nabla^2 w \quad (2.67)$$

$$\nabla^2 \psi = -w \quad (2.68)$$

$$\frac{\partial e}{\partial t} + \mathbf{c} \cdot \nabla e = \frac{1}{ReSc} \nabla^2 e \quad (2.69)$$

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 (1964) **reference**, the ψ is constant along a streamline, then the streamfunction boundary condition can be calculated by $\psi_2 - \psi_1 = \int (udy - vdx)$, 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, $\psi_2 = \int (u_o dy - v_o dx)$.

One of the main difficulties for Streamfunction-Vorticity Formulation is due to absence of vorticity boundary condition, as shown in **references**. In this work, the vorticity boundary condition was calculated by $\omega = \text{curl } \mathbf{v}$ at each time step.

3 FINITE ELEMENT METHOD

3.1 Introduction

In this chapter, we will describe the Finite Element Method (FEM). While the Finite Difference Method (MDF) represents a direct approximation to the differential equations, the proposal of the finite element procedure is an approximation applied to the terms of the variational formulation as mentioned by Zienkiewicz and Cheung (1965) [31]. For more details on the Finite Element Method see the works of Zienkiewicz and Taylor (2000) [48], Hughes (2000) [49] and Fish and Belytschko (2007) [50].

First, we will discretize the governing equations in time using the Taylor series approximation, keeping the second order terms of the expansion in order to reduce the spurious oscillations present in the convection-diffusion equations as in the case of the vorticity and concentration equations. Then, we will present the strong formulation of them. That done, the weak formulation of the governing equations is presented and they are discretized in space using Galerkin's Method with a linear triangular element. Therefore, the government equations in the matrix form according to the *Taylor-Galerkin* Method are presented. For more details on the scheme, see the work of Donea (1984) [37].

3.2 Strong Formulation

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

$$\dot{w} + \mathbf{v} \cdot \nabla w = \frac{1}{Re} \nabla^2 w + \frac{\Delta t}{2} \mathbf{v} \cdot \nabla [\mathbf{v} \cdot \nabla w] \quad (3.1)$$

$$\nabla^2 \psi = -w \quad (3.2)$$

$$\mathbf{v} = \mathbf{D}\psi \quad (3.3)$$

$$\dot{c} + \mathbf{v} \cdot \nabla c = \frac{1}{ReSc} \nabla^2 c + \frac{\Delta t}{2} \mathbf{v} \cdot \nabla [\mathbf{v} \cdot \nabla c] \quad (3.4)$$

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

$$\begin{aligned} w &= w_\Gamma \quad \text{in } \Gamma_1 \\ \psi &= \psi_\Gamma \quad \text{in } \Gamma_2 \\ \mathbf{v} &= \mathbf{v}_\Gamma \quad \text{in } \Gamma_3 \\ c &= c_\Gamma \quad \text{in } \Gamma_4 \end{aligned} \tag{3.5}$$

3.3 Weak Formulation

The result of the weighted governing equations over the domain is known as **Weak formulation** as mentioned by Anjos (2007) [51]. Next, we will present the weak formulation for a single-phase, Newtonian and incompressible fluid using the vorticity-streamfunction formulation with species transport equation. For more details, see the work of Brenner and Scott (1994) [52]. 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:

$$\dot{w} + \mathbf{v} \cdot \nabla w - \frac{1}{Re} \nabla^2 w - \frac{\Delta t}{2} \mathbf{v} \cdot \nabla [\mathbf{v} \cdot \nabla w] = R_1 \tag{3.6}$$

$$\nabla^2 \psi + w = R_2 \tag{3.7}$$

$$\mathbf{v} - \mathbf{D}\psi = R_3 \tag{3.8}$$

$$\dot{c} + \mathbf{v} \cdot \nabla c - \frac{1}{ReSc} \nabla^2 c - \frac{\Delta t}{2} \mathbf{v} \cdot \nabla [\mathbf{v} \cdot \nabla c] = R_4 \tag{3.9}$$

We will try to force the residue to be equivalent to zero in a weighted sense as mentioned by Finlayson (1972) [53], then:

$$\int_{\Omega} R_1 \cdot \delta d\Omega = 0 \quad (3.10)$$

$$\int_{\Omega} R_2 \cdot \phi d\Omega = 0 \quad (3.11)$$

$$\int_{\Omega} R_3 \cdot \xi d\Omega = 0 \quad (3.12)$$

$$\int_{\Omega} R_4 \cdot \eta d\Omega = 0 \quad (3.13)$$

where δ, ϕ, ξ e 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:

$$\int_{\Omega} \left\{ \dot{w} + \mathbf{v} \cdot \nabla w - \frac{1}{Re} \nabla^2 w - \frac{\Delta t}{2} \mathbf{v} \cdot \nabla [\mathbf{v} \cdot \nabla w] \right\} \cdot \delta d\Omega = 0 \quad (3.14)$$

$$\int_{\Omega} \{\nabla^2 \psi + w\} \cdot \phi d\Omega = 0 \quad (3.15)$$

$$\int_{\Omega} \{\mathbf{v} - \mathbf{D}\psi\} \cdot \xi d\Omega = 0 \quad (3.16)$$

$$\int_{\Omega} \left\{ \dot{c} + \mathbf{v} \cdot \nabla c - \frac{1}{ReSc} \nabla^2 c - \frac{\Delta t}{2} \mathbf{v} \cdot \nabla [\mathbf{v} \cdot \nabla c] \right\} \cdot \eta d\Omega = 0 \quad (3.17)$$

Developing the integrals, we have:

$$\int_{\Omega} \dot{w} \delta d\Omega + \int_{\Omega} \mathbf{v} \cdot \nabla w \delta d\Omega - \frac{1}{Re} \int_{\Omega} \nabla^2 w \delta - \frac{\Delta t}{2} \int_{\Omega} \mathbf{v} \cdot \nabla [\mathbf{v} \cdot \nabla w] \delta d\Omega = 0 \quad (3.18)$$

$$\int_{\Omega} \nabla^2 \psi \phi d\Omega + \int_{\Omega} w \phi d\Omega = 0 \quad (3.19)$$

$$\int_{\Omega} \mathbf{v} \cdot \nabla \xi d\Omega - \int_{\Omega} \mathbf{D} \psi \cdot \nabla \xi d\Omega = 0 \quad (3.20)$$

$$\int_{\Omega} \dot{c} \eta d\Omega + \int_{\Omega} \mathbf{v} \cdot \nabla c \eta d\Omega - \frac{1}{ReSc} \int_{\Omega} \nabla^2 c \eta d\Omega - \frac{\Delta t}{2} \int_{\Omega} \mathbf{v} \cdot \nabla [\mathbf{v} \cdot \nabla c] \eta d\Omega = 0 \quad (3.21)$$

In the diffusive term of the equations (3.18, 3.19 and 3.21), 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:

$$-\frac{1}{Re} \int_{\Omega} \nabla^2 w \delta d\Omega = \frac{1}{Re} \int_{\Omega} \nabla w \cdot \nabla \delta d\Omega - \frac{1}{Re} \int_{\Gamma} \delta \nabla w \cdot \mathbf{n} d\Gamma \quad (3.22)$$

$$\int_{\Omega} \nabla^2 \psi \phi d\Omega = - \int_{\Omega} \nabla \psi \cdot \nabla \phi d\Omega + \int_{\Gamma} \phi \nabla \psi \cdot \mathbf{n} d\Gamma \quad (3.23)$$

$$-\frac{1}{ReSc} \int_{\Omega} \nabla^2 c \eta d\Omega = \frac{1}{ReSc} \int_{\Omega} \nabla c \cdot \nabla \eta d\Omega - \frac{1}{ReSc} \int_{\Gamma} \eta \nabla c \cdot \mathbf{n} d\Gamma \quad (3.24)$$

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 presented in the section 3.2, for the problem proposed in this work, we have only Dirichlet conditions (known as an essential condition).

Therefore, we will consider $\delta = 0$, $\phi = 0$ and $\eta = 0$ assumptions in the equations (3.22, 3.23 and 3.24) for all Γ . Thereby, the integral in Γ will be null and the diffusive term of the equations (3.18, 3.19 and 3.21) become:

$$-\frac{1}{Re} \int_{\Omega} \nabla^2 w \delta d\Omega = \frac{1}{Re} \int_{\Omega} \nabla w \cdot \nabla \delta d\Omega \quad (3.25)$$

$$\int_{\Omega} \nabla^2 \psi \phi d\Omega = - \int_{\Omega} \nabla \psi \cdot \nabla \phi d\Omega \quad (3.26)$$

$$-\frac{1}{ReSc} \int_{\Omega} \nabla^2 c \eta d\Omega = \frac{1}{ReSc} \int_{\Omega} \nabla c \cdot \nabla \eta d\Omega \quad (3.27)$$

For the numerical diffusivity term of the equations (3.18 and 3.21), we will apply a procedure similar to the previous one. Thus, the term of numerical diffusivity will be:

$$-\frac{\Delta t}{2} \int_{\Omega} \mathbf{v} \cdot \nabla [\mathbf{v} \cdot \nabla w] \delta d\Omega = \frac{\Delta t}{2} \int_{\Omega} [\mathbf{v} \cdot \nabla w] \mathbf{v} \cdot \nabla \delta d\Omega - \frac{\Delta t}{2} \int_{\Gamma} [\mathbf{v} \cdot \nabla w] \delta \mathbf{v} \cdot \mathbf{n} d\Gamma \quad (3.28)$$

$$-\frac{\Delta t}{2} \int_{\Omega} \mathbf{v} \cdot \nabla [\mathbf{v} \cdot \nabla c] \eta d\Omega = \frac{\Delta t}{2} \int_{\Omega} [\mathbf{v} \cdot \nabla c] \mathbf{v} \cdot \nabla \eta d\Omega - \frac{\Delta t}{2} \int_{\Gamma} [\mathbf{v} \cdot \nabla c] \eta \mathbf{v} \cdot \mathbf{n} d\Gamma \quad (3.29)$$

In the same way, we can cancel the natural condition term according to the assumption previously adopted. Therefore, the numeric diffusivity term for the equations (3.18 and 3.21) will be:

$$-\frac{\Delta t}{2} \int_{\Omega} \mathbf{v} \cdot \nabla [\mathbf{v} \cdot \nabla w] \delta d\Omega = \frac{\Delta t}{2} \int_{\Omega} [\mathbf{v} \cdot \nabla w] \mathbf{v} \cdot \nabla \delta d\Omega \quad (3.30)$$

$$-\frac{\Delta t}{2} \int_{\Omega} \mathbf{v} \cdot \nabla [\mathbf{v} \cdot \nabla c] \eta d\Omega = \frac{\Delta t}{2} \int_{\Omega} [\mathbf{v} \cdot \nabla c] \mathbf{v} \cdot \nabla \eta d\Omega \quad (3.31)$$

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

$$\int_{\Omega} \dot{w} \delta d\Omega + \int_{\Omega} \mathbf{v} \cdot \nabla w \delta d\Omega + \frac{1}{Re} \int_{\Omega} \nabla w \cdot \nabla \delta d\Omega + \frac{\Delta t}{2} \int_{\Omega} [\mathbf{v} \cdot \nabla w] \mathbf{v} \cdot \nabla \delta d\Omega = 0 \quad (3.32)$$

$$-\int_{\Omega} \nabla \psi \cdot \nabla \phi d\Omega + \int_{\Omega} w \phi d\Omega = 0 \quad (3.33)$$

$$\int_{\Omega} \mathbf{v} \xi d\Omega - \int_{\Omega} \mathbf{D} \psi \xi d\Omega = 0 \quad (3.34)$$

$$\int_{\Omega} \dot{c} \eta d\Omega + \int_{\Omega} \mathbf{v} \cdot \nabla c \eta d\Omega + \frac{1}{ReSc} \int_{\Omega} \nabla c \cdot \nabla \eta d\Omega + \frac{\Delta t}{2} \int_{\Omega} [\mathbf{v} \cdot \nabla c] \mathbf{v} \cdot \nabla \eta d\Omega = 0 \quad (3.35)$$

If we consider that:

$$\begin{aligned}
 m_1(\dot{w}, \delta) &= \int_{\Omega} \dot{w} \delta d\Omega & m_3(\mathbf{v}, \xi) &= \int_{\Omega} \mathbf{v} \cdot \nabla \xi d\Omega \\
 g_1(\mathbf{v}, \delta) &= \int_{\Omega} \mathbf{v} \cdot \nabla w \delta d\Omega & g_3(\psi, \xi) &= \int_{\Omega} \mathbf{D}\psi \cdot \nabla \xi d\Omega \\
 k_1(w, \delta) &= \int_{\Omega} \nabla w \cdot \nabla \delta d\Omega & m_4(\dot{c}, \eta) &= \int_{\Omega} \dot{c} \eta d\Omega \\
 k_{n1}(w, \delta) &= \int_{\Omega} [\mathbf{v} \cdot \nabla w] \mathbf{v} \cdot \nabla \delta d\Omega & g_4(\mathbf{v}, \eta) &= \int_{\Omega} \mathbf{v} \cdot \nabla c \eta d\Omega \\
 k_2(\psi, \phi) &= \int_{\Omega} \nabla \psi \cdot \nabla \phi d\Omega & k_4(c, \eta) &= \int_{\Omega} \nabla c \cdot \nabla \eta d\Omega \\
 m_2(\psi, \phi) &= \int_{\Omega} w \phi d\Omega & k_{n4}(c, \delta) &= \int_{\Omega} [\mathbf{v} \cdot \nabla c] \mathbf{v} \cdot \nabla \eta d\Omega
 \end{aligned} \tag{3.36}$$

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

$$m_1(\dot{w}, \delta) + g_1(\mathbf{v}, \delta) + \frac{1}{Re} k_1(w, \delta) + \frac{\Delta t}{2} k_{n1}(w, \delta) = 0 \tag{3.37}$$

$$-k_2(\psi, \phi) + m_2(\psi, \phi) = 0 \tag{3.38}$$

$$m_3(\mathbf{v}, \xi) - g_3(\psi, \xi) = 0 \tag{3.39}$$

$$m_4(\dot{c}, \eta) + g_4(\mathbf{v}, \eta) + \frac{1}{ReSc} k_4(c, \eta) + \frac{\Delta t}{2} k_{n4}(c, \eta) = 0 \tag{3.40}$$

Considering the sets of basis functions:

$$\begin{aligned}
 \mathbb{W} &= \{w \in \Omega \rightarrow \mathbb{R}^2 : \int_{\Omega} w^2 d\Omega < \infty; w = w_{\Gamma}\} \\
 \mathbb{P} &= \{\psi \in \Omega \rightarrow \mathbb{R}^2 : \int_{\Omega} \psi^2 d\Omega < \infty; \psi = \psi_{\Gamma}\} \\
 \mathbb{V} &= \{v \in \Omega \rightarrow \mathbb{R}^2 : \int_{\Omega} v^2 d\Omega < \infty; v = v_{\Gamma}\} \\
 \mathbb{C} &= \{c \in \Omega \rightarrow \mathbb{R}^2 : \int_{\Omega} c^2 d\Omega < \infty; c = c_{\Gamma}\}
 \end{aligned} \tag{3.41}$$

and the set of weight functions space:

$$\begin{aligned}\mathbb{D} &= \{\delta \in \Omega \rightarrow \mathbb{R}^2 : \int_{\Omega} \delta^2 d\Omega < \infty; \delta_{\Gamma} = 0\} \\ \mathbb{F} &= \{\phi \in \Omega \rightarrow \mathbb{R}^2 : \int_{\Omega} \phi^2 d\Omega < \infty; \phi_{\Gamma} = 0\} \\ \mathbb{X} &= \{\xi \in \Omega \rightarrow \mathbb{R}^2 : \int_{\Omega} \xi^2 d\Omega < \infty; \xi_{\Gamma} = 0\} \\ \mathbb{N} &= \{\eta \in \Omega \rightarrow \mathbb{R}^2 : \int_{\Omega} \eta^2 d\Omega < \infty; \eta_{\Gamma} = 0\}\end{aligned}\tag{3.42}$$

The weak formulation consists to find the solutions of $w \in \mathbb{W}$, $\psi \in \mathbb{P}$, $v \in \mathbb{V}$ and $c \in \mathbb{C}$ such that:

$$m_1(\dot{w}, \delta) + g_1(\mathbf{v}, \delta) + \frac{1}{Re} k_1(w, \delta) + \frac{\Delta t}{2} k_{n1}(w, \delta) = 0\tag{3.43}$$

$$-k_2(\psi, \phi) + m_2(\psi, \phi) = 0\tag{3.44}$$

$$m_3(\mathbf{v}, \xi) - g_3(\psi, \xi) = 0\tag{3.45}$$

$$m_4(\dot{c}, \eta) + g_4(\mathbf{v}, \eta) + \frac{1}{ReSc} k_4(c, \eta) + \frac{\Delta t}{2} k_{n4}(c, \eta) = 0\tag{3.46}$$

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

3.4 Galerkin Method

The weight functions δ , ϕ , ξ 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 , \mathbf{v} , c). Thus, Eqs. 3.32 to 3.35 can be presented in expanded form such as:

$$\begin{aligned}\int_{\Omega} \dot{w} \delta d\Omega + \int_{\Omega} u \frac{\partial w}{\partial x} \delta d\Omega + \int_{\Omega} v \frac{\partial w}{\partial y} \delta d\Omega + \frac{1}{Re} \int_{\Omega} \left\{ \frac{\partial w}{\partial x} \frac{\partial \delta}{\partial x} + \frac{\partial w}{\partial y} \frac{\partial \delta}{\partial y} \right\} d\Omega \\ + \frac{\Delta t}{2} \int_{\Omega} u \frac{\partial \delta}{\partial x} \left[u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} \right] d\Omega + \frac{\Delta t}{2} \int_{\Omega} v \frac{\partial \delta}{\partial y} \left[u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} \right] d\Omega = 0\end{aligned}\tag{3.47}$$

$$-\int_{\Omega} \left\{ \frac{\partial \psi}{\partial x} \frac{\partial \phi}{\partial x} + \frac{\partial \psi}{\partial y} \frac{\partial \phi}{\partial y} \right\} d\Omega + \int_{\Omega} w \phi d\Omega = 0\tag{3.48}$$

$$\int_{\Omega} u \xi d\Omega - \int_{\Omega} \frac{\partial \psi}{\partial y} \xi d\Omega = 0 \quad (3.49)$$

$$\int_{\Omega} v \xi d\Omega + \int_{\Omega} \frac{\partial \psi}{\partial x} \xi d\Omega = 0 \quad (3.50)$$

$$\begin{aligned} & \int_{\Omega} \dot{c} \eta d\Omega + \int_{\Omega} u \frac{\partial c}{\partial x} \eta d\Omega + \int_{\Omega} v \frac{\partial c}{\partial y} \eta d\Omega + \frac{1}{ReSc} \int_{\Omega} \left\{ \frac{\partial c}{\partial x} \frac{\partial \eta}{\partial x} + \frac{\partial c}{\partial y} \frac{\partial \eta}{\partial y} \right\} d\Omega \\ & + \frac{\Delta t}{2} \int_{\Omega} u \frac{\partial \eta}{\partial x} \left[u \frac{\partial c}{\partial x} + v \frac{\partial c}{\partial y} \right] d\Omega + \frac{\Delta t}{2} \int_{\Omega} v \frac{\partial \eta}{\partial y} \left[u \frac{\partial c}{\partial x} + v \frac{\partial c}{\partial y} \right] d\Omega = 0 \end{aligned} \quad (3.51)$$

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:

$$w(\mathbf{x}, t) \simeq \sum_{i=1}^{np} w_i(t) N_i(\mathbf{x}) \quad (3.52)$$

$$\psi(\mathbf{x}, t) \simeq \sum_{i=1}^{np} \psi_i(t) N_i(\mathbf{x}) \quad (3.53)$$

$$u(\mathbf{x}, t) \simeq \sum_{i=1}^{np} u_i(t) N_i(\mathbf{x}) \quad (3.54)$$

$$v(\mathbf{x}, t) \simeq \sum_{i=1}^{np} v_i(t) N_i(\mathbf{x}) \quad (3.55)$$

$$c(\mathbf{x}, t) \simeq \sum_{i=1}^{np} c_i(t) N_i(\mathbf{x}) \quad (3.56)$$

where $w_i = [w_1, \dots, w_{np}]$, $\psi_i = [\psi_1, \dots, \psi_{np}]$, $u_i = [u_1, \dots, u_{np}]$, $v_i = [v_1, \dots, v_{np}]$, $c_i = [c_1, \dots, c_{np}]$ are the unknowns to be found. As these unknowns depend only on time, they can leave the integrals on the domain Ω , whereas $N_i = [N_1, \dots, N_{np}]$ are the approximation functions known as basis functions or interpolation functions. These functions can be chosen arbitrarily and 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:

$$\delta(\mathbf{x}, t) \simeq \sum_{j=1}^{np} \delta_j(t) N_j(\mathbf{x}) \quad (3.57)$$

$$\phi(\mathbf{x}, t) \simeq \sum_{j=1}^{np} \phi_j(t) N_j(\mathbf{x}) \quad (3.58)$$

$$\xi(\mathbf{x}, t) \simeq \sum_{j=1}^{np} \xi_j(t) N_j(\mathbf{x}) \quad (3.59)$$

$$\eta(\mathbf{x}, t) \simeq \sum_{j=1}^{np} \eta_j(t) N_j(\mathbf{x}) \quad (3.60)$$

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

$$\begin{aligned} & \int_{\Omega} \sum_{i=1}^{np} \dot{w}_i N_i \sum_{j=1}^{np} \delta_j N_j d\Omega + u \int_{\Omega} \sum_{i=1}^{np} \frac{\partial w_i N_i}{\partial x} \sum_{j=1}^{np} \delta_j N_j d\Omega + v \int_{\Omega} \sum_{i=1}^{np} \frac{\partial w_i N_i}{\partial y} \sum_{j=1}^{np} \delta_j N_j d\Omega \\ & + \frac{1}{Re} \int_{\Omega} \left\{ \sum_{i=1}^{np} \frac{\partial w_i N_i}{\partial x} \sum_{j=1}^{np} \frac{\partial \delta_j N_j}{\partial x} + \sum_{i=1}^{np} \frac{\partial w_i N_i}{\partial y} \sum_{j=1}^{np} \frac{\partial \delta_j N_j}{\partial y} \right\} \\ & + \frac{\Delta t}{2} \int_{\Omega} u \sum_{j=1}^{np} \frac{\partial \delta_j N_j}{\partial x} \left[u \sum_{i=1}^{np} \frac{\partial w_i N_i}{\partial x} + v \sum_{i=1}^{np} \frac{\partial w_i N_i}{\partial y} \right] d\Omega \\ & + \frac{\Delta t}{2} \int_{\Omega} v \sum_{j=1}^{np} \frac{\partial \delta_j N_j}{\partial y} \left[u \sum_{i=1}^{np} \frac{\partial w_i N_i}{\partial x} + v \sum_{i=1}^{np} \frac{\partial w_i N_i}{\partial y} \right] d\Omega = 0 \end{aligned} \quad (3.61)$$

$$\begin{aligned} & - \int_{\Omega} \left\{ \sum_{i=1}^{np} \frac{\partial \psi_i N_i}{\partial x} \sum_{j=1}^{np} \frac{\partial \phi_j N_j}{\partial x} + \sum_{i=1}^{np} \frac{\partial \psi_i N_i}{\partial y} \sum_{j=1}^{np} \frac{\partial \phi_j N_j}{\partial y} \right\} d\Omega \\ & + \int_{\Omega} \sum_{i=1}^{np} w_i N_i \sum_{j=1}^{np} \phi_j N_j d\Omega = 0 \end{aligned} \quad (3.62)$$

$$\int_{\Omega} \sum_{i=1}^{np} u_i N_i \sum_{j=1}^{np} \xi_j N_j d\Omega - \int_{\Omega} \sum_{i=1}^{np} \frac{\partial \psi_i N_i}{\partial y} \sum_{j=1}^{np} \xi_j N_j d\Omega = 0 \quad (3.63)$$

$$\int_{\Omega} \sum_{i=1}^{np} v_i N_i \sum_{j=1}^{np} \xi_j N_j d\Omega + \int_{\Omega} \sum_{i=1}^{np} \frac{\partial \psi_i N_i}{\partial x} \sum_{j=1}^{np} \xi_j N_j d\Omega = 0 \quad (3.64)$$

$$\begin{aligned} & \int_{\Omega} \sum_{i=1}^{np} \dot{c}_i N_i \sum_{j=1}^{np} \eta_j N_j d\Omega + u \int_{\Omega} \sum_{i=1}^{np} \frac{\partial c_i N_i}{\partial x} \sum_{j=1}^{np} \eta_j N_j d\Omega + v \int_{\Omega} \sum_{i=1}^{np} \frac{\partial c_i N_i}{\partial y} \sum_{j=1}^{np} \eta_j N_j d\Omega \\ & + \frac{1}{ReSc} \int_{\Omega} \left\{ \sum_{i=1}^{np} \frac{\partial c_i N_i}{\partial x} \sum_{j=1}^{np} \frac{\partial \eta_j N_j}{\partial x} + \sum_{i=1}^{np} \frac{\partial c_i N_i}{\partial y} \sum_{j=1}^{np} \frac{\partial \eta_j N_j}{\partial y} \right\} d\Omega \\ & + \frac{\Delta t}{2} \int_{\Omega} u \sum_{j=1}^{np} \frac{\partial \eta_j N_j}{\partial x} \left[u \sum_{i=1}^{np} \frac{\partial c_i N_i}{\partial x} + v \sum_{i=1}^{np} \frac{\partial c_i N_i}{\partial y} \right] d\Omega \\ & + \frac{\Delta t}{2} \int_{\Omega} v \sum_{j=1}^{np} \frac{\partial \eta_j N_j}{\partial y} \left[u \sum_{i=1}^{np} \frac{\partial c_i N_i}{\partial x} + v \sum_{i=1}^{np} \frac{\partial c_i N_i}{\partial y} \right] d\Omega = 0 \end{aligned} \quad (3.65)$$

We can remove the velocity components u and v from the integral in the convective term of the equations 3.61 and 3.65 since they will not be treated as unknowns, that is, for the vorticity equation we will use the values of the u and v components of the previous time step while for the species transport equation, the velocity components are calculated firstly. In this way, these equations are transformed into linear equations. Moving the sum symbols out of integrals, we have:

$$\begin{aligned} & \sum_{j=1}^{np} \delta_j \left[\sum_{i=1}^{np} \dot{w}_i \int_{\Omega} N_i N_j d\Omega + \sum_{i=1}^{np} w_i \left[u \int_{\Omega} \frac{\partial N_i}{\partial x} N_j d\Omega + v \int_{\Omega} \frac{\partial N_i}{\partial y} N_j d\Omega \right. \right. \\ & + \frac{1}{Re} \int_{\Omega} \left\{ \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right\} d\Omega + \frac{\Delta t}{2} \int_{\Omega} u \frac{\partial N_j}{\partial x} \left[u \frac{\partial N_i}{\partial x} + v \frac{\partial N_i}{\partial y} \right] d\Omega \\ & \left. \left. + \frac{\Delta t}{2} \int_{\Omega} v \frac{\partial N_j}{\partial y} \left[u \frac{\partial N_i}{\partial x} + v \frac{\partial N_i}{\partial y} \right] d\Omega \right] \right] = 0 \end{aligned} \quad (3.66)$$

$$\sum_{j=1}^{np} \phi_j \left[\sum_{i=1}^{np} \psi_i \left[- \int_{\Omega} \left\{ \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right\} d\Omega + w_i \int_{\Omega} N_i N_j d\Omega \right] \right] = 0 \quad (3.67)$$

$$\sum_{j=1}^{np} \xi_j \left[\sum_{i=1}^{np} u_i \left[\int_{\Omega} N_i N_j d\Omega - \psi_i \int_{\Omega} \frac{\partial N_i}{\partial y} N_j d\Omega \right] \right] = 0 \quad (3.68)$$

$$\sum_{j=1}^{np} \xi_j \left[\sum_{i=1}^{np} v_i \left[\int_{\Omega} N_i N_j d\Omega + \psi_i \int_{\Omega} \frac{\partial N_i}{\partial x} N_j d\Omega \right] \right] = 0 \quad (3.69)$$

$$\begin{aligned} & \sum_{j=1}^{np} \eta_j \left[\sum_{i=1}^{np} \dot{c}_i \int_{\Omega} N_i N_j d\Omega + \sum_{i=1}^{np} c_i \left[u \int_{\Omega} \frac{\partial N_i}{\partial x} N_j d\Omega + v \int_{\Omega} \frac{\partial N_i}{\partial y} N_j d\Omega \right. \right. \\ & + \frac{1}{ReSc} \int_{\Omega} \left\{ \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right\} d\Omega + \frac{\Delta t}{2} \int_{\Omega} u \frac{\partial N_j}{\partial x} \left[u \frac{\partial N_i}{\partial x} + v \frac{\partial N_i}{\partial y} \right] d\Omega \\ & \left. \left. + \frac{\Delta t}{2} \int_{\Omega} v \frac{\partial N_j}{\partial y} \left[u \frac{\partial N_i}{\partial x} + v \frac{\partial N_i}{\partial y} \right] d\Omega \right] \right] = 0 \end{aligned} \quad (3.70)$$

Whereas $\sum_{j=1}^{np} \delta_j \neq 0$, $\sum_{j=1}^{np} \phi_j \neq 0$, $\sum_{j=1}^{np} \xi_j \neq 0$ and $\sum_{j=1}^{np} \eta_j \neq 0$, thus the governing equations become:

$$\begin{aligned} & \sum_{j=1}^{np} \sum_{i=1}^{np} \dot{w}_i \int_{\Omega} N_i N_j d\Omega + \sum_{j=1}^{np} \sum_{i=1}^{np} w_i \left[u \int_{\Omega} \frac{\partial N_i}{\partial x} N_j d\Omega + v \int_{\Omega} \frac{\partial N_i}{\partial y} N_j d\Omega \right. \\ & + \frac{1}{Re} \int_{\Omega} \left\{ \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right\} d\Omega + \frac{\Delta t}{2} \int_{\Omega} u \frac{\partial N_j}{\partial x} \left[u \frac{\partial N_i}{\partial x} + v \frac{\partial N_i}{\partial y} \right] d\Omega \\ & \left. + \frac{\Delta t}{2} \int_{\Omega} v \frac{\partial N_j}{\partial y} \left[u \frac{\partial N_i}{\partial x} + v \frac{\partial N_i}{\partial y} \right] d\Omega \right] = 0 \end{aligned} \quad (3.71)$$

$$\sum_{j=1}^{np} \sum_{i=1}^{np} \psi_i \left[- \int_{\Omega} \left\{ \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right\} d\Omega + w_i \int_{\Omega} N_i N_j d\Omega \right] = 0 \quad (3.72)$$

$$\sum_{j=1}^{np} \sum_{i=1}^{np} u_i \left[\int_{\Omega} N_i N_j d\Omega - \psi_i \int_{\Omega} \frac{\partial N_i}{\partial y} N_j d\Omega \right] = 0 \quad (3.73)$$

$$\sum_{j=1}^{np} \sum_{i=1}^{np} v_i \left[\int_{\Omega} N_i N_j d\Omega + \psi_i \int_{\Omega} \frac{\partial N_i}{\partial x} N_j d\Omega \right] = 0 \quad (3.74)$$

$$\begin{aligned} & \sum_{j=1}^{np} \sum_{i=1}^{np} c_i \int_{\Omega} N_i N_j d\Omega + \sum_{j=1}^{np} \sum_{i=1}^{np} c_i \left[u \int_{\Omega} \frac{\partial N_i}{\partial x} N_j d\Omega + v \int_{\Omega} \frac{\partial N_i}{\partial y} N_j d\Omega \right. \\ & + \frac{1}{ReSc} \int_{\Omega} \left\{ \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right\} d\Omega + \frac{\Delta t}{2} \int_{\Omega} u \frac{\partial N_j}{\partial x} \left[u \frac{\partial N_i}{\partial x} + v \frac{\partial N_i}{\partial y} \right] d\Omega \\ & \left. + \frac{\Delta t}{2} \int_{\Omega} v \frac{\partial N_j}{\partial y} \left[u \frac{\partial N_i}{\partial x} + v \frac{\partial N_i}{\partial y} \right] d\Omega \right] = 0 \end{aligned} \quad (3.75)$$

that is:

$$\begin{aligned} & \sum_{j=1}^{np} \sum_{i=1}^{np} \dot{w}_i \int_{\Omega} N_i N_j d\Omega + \sum_{j=1}^{np} \sum_{i=1}^{np} w_i \left[u \int_{\Omega} \frac{\partial N_i}{\partial x} N_j d\Omega + v \int_{\Omega} \frac{\partial N_i}{\partial y} N_j d\Omega \right. \\ & + \frac{1}{Re} \int_{\Omega} \left\{ \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right\} d\Omega + \frac{\Delta t}{2} \int_{\Omega} u \left[u \frac{\partial N_j}{\partial x} \frac{\partial N_i}{\partial x} + v \frac{\partial N_j}{\partial x} \frac{\partial N_i}{\partial y} \right] d\Omega \\ & \left. + \frac{\Delta t}{2} \int_{\Omega} v \left[u \frac{\partial N_j}{\partial y} \frac{\partial N_i}{\partial x} + v \frac{\partial N_j}{\partial y} \frac{\partial N_i}{\partial y} \right] d\Omega \right] = 0 \end{aligned} \quad (3.76)$$

$$\sum_{j=1}^{np} \sum_{i=1}^{np} \psi_i \left[- \int_{\Omega} \left\{ \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right\} d\Omega + w_i \int_{\Omega} N_i N_j d\Omega \right] = 0 \quad (3.77)$$

$$\sum_{j=1}^{np} \sum_{i=1}^{np} u_i \left[\int_{\Omega} N_i N_j d\Omega - \psi_i \int_{\Omega} \frac{\partial N_i}{\partial y} N_j d\Omega \right] = 0 \quad (3.78)$$

$$\sum_{j=1}^{np} \sum_{i=1}^{np} v_i \left[\int_{\Omega} N_i N_j d\Omega + \psi_i \int_{\Omega} \frac{\partial N_i}{\partial x} N_j d\Omega \right] = 0 \quad (3.79)$$

$$\begin{aligned}
& \sum_{j=1}^{np} \sum_{i=1}^{np} \dot{c}_i \int_{\Omega} N_i N_j d\Omega + \sum_{j=1}^{np} \sum_{i=1}^{np} c_i \left[u \int_{\Omega} \frac{\partial N_i}{\partial x} N_j d\Omega + v \int_{\Omega} \frac{\partial N_i}{\partial y} N_j d\Omega \right. \\
& + \frac{1}{ReSc} \int_{\Omega} \left\{ \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right\} d\Omega + \frac{\Delta t}{2} \int_{\Omega} u \left[u \frac{\partial N_j}{\partial x} \frac{\partial N_i}{\partial x} + v \frac{\partial N_j}{\partial x} \frac{\partial N_i}{\partial y} \right] d\Omega \\
& \left. + \frac{\Delta t}{2} \int_{\Omega} v \left[u \frac{\partial N_j}{\partial y} \frac{\partial N_i}{\partial x} + v \frac{\partial N_j}{\partial y} \frac{\partial N_i}{\partial y} \right] d\Omega \right] = 0
\end{aligned} \quad (3.80)$$

3.5 Semi-Lagrangian Method

Nesta seção discretizaremos as equações de governo no tempo, através da expansão em série de Taylor para a variável em questão a fim de aproximar a derivada temporal. Com o intuito de simplificação, apresentaremos a discretização da equação da vorticidade. Um procedimento semelhante poderá ser feito para a equação de transporte de espécie química (Eq. 2.69). Sendo assim, expandindo os termos da equação da vorticidade (Eq. 2.67), temos:

$$\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} = \frac{1}{Re} \frac{\partial^2 w}{\partial x^2} + \frac{1}{Re} \frac{\partial^2 w}{\partial y^2} \quad (3.81)$$

isto é:

$$\frac{\partial w}{\partial t} = -u \frac{\partial w}{\partial x} - v \frac{\partial w}{\partial y} + \frac{1}{Re} \frac{\partial^2 w}{\partial x^2} + \frac{1}{Re} \frac{\partial^2 w}{\partial y^2} \quad (3.82)$$

Multiplicando ambos os lado por $\partial/\partial t$, temos:

$$\frac{\partial}{\partial t} \left[\frac{\partial w}{\partial t} \right] = \frac{\partial}{\partial t} \left[-u \frac{\partial w}{\partial x} - v \frac{\partial w}{\partial y} + \frac{1}{Re} \frac{\partial^2 w}{\partial x^2} + \frac{1}{Re} \frac{\partial^2 w}{\partial y^2} \right] \quad (3.83)$$

Desta forma, considerando a expansão de Taylor:

$$w^{n+1} = \sum_{k=0}^{\infty} \frac{\partial^k w^n}{\partial t^k} \frac{\Delta t^k}{k!} \quad (3.84)$$

Desenvolvendo a série, temos:

$$w^{n+1} = w^n + \frac{\partial w^n}{\partial t} \frac{\Delta t}{1!} + \frac{\partial^2 w^n}{\partial t^2} \frac{\Delta t^2}{2!} + \frac{\partial^3 w^n}{\partial t^3} \frac{\Delta t^3}{3!} + \dots \quad (3.85)$$

Caso os termos de ordem superior a dois forem omitidos, a equação fica da forma:

$$w^{n+1} = w^n + \frac{\partial w^n}{\partial t} \Delta t + \frac{\partial^2 w^n}{\partial t^2} \frac{\Delta t^2}{2} + O(\Delta t^3) \quad (3.86)$$

onde $O(\Delta t^3)$ é o erro devido ao truncamento da série. Graficamente, esta aproximação pode ser representada como apresentado na Figure 3, a seguir:

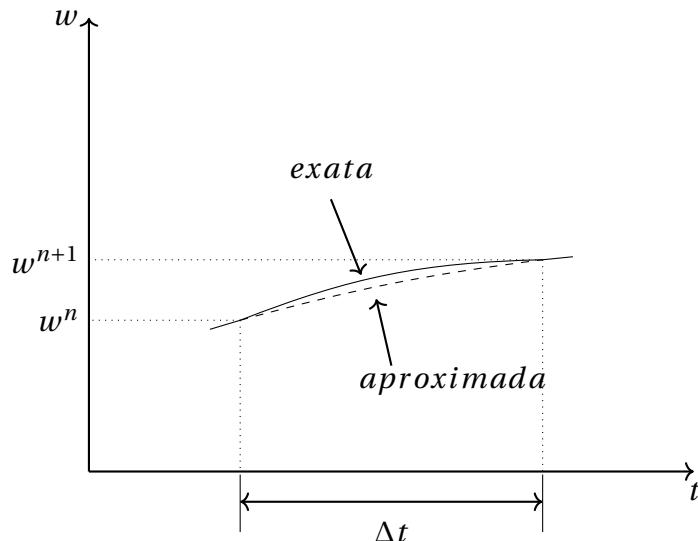


Figure 3: Variação da vorticidade em um passo de tempo

Omitindo o erro do truncamento, a derivada temporal (Eq. 3.86) pode ser aproximada a:

$$w^{n+1} = w^n + \frac{\partial w^n}{\partial t} \Delta t + \frac{\partial^2 w^n}{\partial t^2} \frac{\Delta t^2}{2} \quad (3.87)$$

onde w^{n+1} é a vorticidade que será calculada e w^n é a vorticidade que foi calculada no passo de tempo anterior. Substituindo a equação da vorticidade (Eq. 3.82) e a equação da vorticidade multiplicada pela derivada temporal (Eq. 3.83) na aproximação feita acima,

temos:

$$\begin{aligned} w^{n+1} = w^n + \Delta t & \left[-u \frac{\partial w^n}{\partial x} - v \frac{\partial w^n}{\partial y} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial x^2} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial y^2} \right] \\ & + \frac{\Delta t^2}{2} \left[\frac{\partial}{\partial t} \left[-u \frac{\partial w^n}{\partial x} - v \frac{\partial w^n}{\partial y} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial x^2} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial y^2} \right] \right] \end{aligned} \quad (3.88)$$

Assumindo que u e v são constantes, temos:

$$\begin{aligned} w^{n+1} = w^n + \Delta t & \left[-u \frac{\partial w^n}{\partial x} - v \frac{\partial w^n}{\partial y} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial x^2} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial y^2} \right] \\ & + \frac{\Delta t^2}{2} \left[-u \frac{\partial}{\partial t} \frac{\partial w^n}{\partial x} - v \frac{\partial}{\partial t} \frac{\partial w^n}{\partial y} + \frac{1}{Re} \frac{\partial}{\partial t} \frac{\partial^2 w^n}{\partial x^2} + \frac{1}{Re} \frac{\partial}{\partial t} \frac{\partial^2 w^n}{\partial y^2} \right] \end{aligned} \quad (3.89)$$

Invertendo as ordens de derivação do último termo, encontramos:

$$\begin{aligned} w^{n+1} = w^n + \Delta t & \left[-u \frac{\partial w^n}{\partial x} - v \frac{\partial w^n}{\partial x} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial x^2} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial y^2} \right] \\ & + \frac{\Delta t^2}{2} \left[-u \frac{\partial}{\partial x} \frac{\partial w^n}{\partial t} - v \frac{\partial}{\partial y} \frac{\partial w^n}{\partial t} + \frac{1}{Re} \frac{\partial^2}{\partial x^2} \frac{\partial w^n}{\partial t} + \frac{1}{Re} \frac{\partial^2}{\partial y^2} \frac{\partial w^n}{\partial t} \right] \end{aligned} \quad (3.90)$$

Substituindo os termo $\partial w / \partial t$ pela equação de vorticidade (Eq. 3.82), temos:

$$\begin{aligned} w^{n+1} = w^n + \Delta t & \left[-u \frac{\partial w^n}{\partial x} - v \frac{\partial w^n}{\partial x} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial x^2} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial y^2} \right] \\ & + \frac{\Delta t^2}{2} \left[-u \frac{\partial}{\partial x} \left[-u \frac{\partial w^n}{\partial x} - v \frac{\partial w^n}{\partial y} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial x^2} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial y^2} \right] \right. \\ & \quad \left. - v \frac{\partial}{\partial y} \left[-u \frac{\partial w^n}{\partial x} - v \frac{\partial w^n}{\partial y} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial x^2} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial y^2} \right] \right. \\ & \quad \left. + \frac{\partial^2}{\partial x^2} \left[-u \frac{\partial w^n}{\partial x} - v \frac{\partial w^n}{\partial y} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial x^2} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial y^2} \right] \right. \\ & \quad \left. + \frac{\partial^2}{\partial y^2} \left[-u \frac{\partial w^n}{\partial x} - v \frac{\partial w^n}{\partial y} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial x^2} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial y^2} \right] \right] \end{aligned} \quad (3.91)$$

Truncando os termos de ordem superior a dois, obtemos:

$$\begin{aligned} w^{n+1} = w^n + \Delta t & \left[-u \frac{\partial w^n}{\partial x} - v \frac{\partial w^n}{\partial x} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial x^2} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial y^2} \right] \\ & + \frac{\Delta t^2}{2} \left[-u \frac{\partial}{\partial x} \left[-u \frac{\partial w^n}{\partial x} - v \frac{\partial w^n}{\partial y} \right] - v \frac{\partial}{\partial y} \left[-u \frac{\partial w^n}{\partial x} - v \frac{\partial w^n}{\partial y} \right] \right] + O(\Delta t^3) \end{aligned} \quad (3.92)$$

Omitindo novamente o erro de truncamento, possuímos a seguinte equação:

$$\begin{aligned} w^{n+1} = w^n + \Delta t & \left[-u \frac{\partial w^n}{\partial x} - v \frac{\partial w^n}{\partial x} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial x^2} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial y^2} \right] \\ & + \frac{\Delta t^2}{2} \left[-u \frac{\partial}{\partial x} \left[-u \frac{\partial w^n}{\partial x} - v \frac{\partial w^n}{\partial y} \right] - v \frac{\partial}{\partial y} \left[-u \frac{\partial w^n}{\partial x} - v \frac{\partial w^n}{\partial y} \right] \right] \end{aligned} \quad (3.93)$$

isto é:

$$\begin{aligned} \left[\frac{w^{n+1} - w^n}{\Delta t} \right] + u \frac{\partial w^n}{\partial x} + v \frac{\partial w^n}{\partial y} &= \frac{1}{Re} \frac{\partial^2 w^n}{\partial x^2} + \frac{1}{Re} \frac{\partial^2 w^n}{\partial y^2} \\ & + \frac{\Delta t}{2} u \frac{\partial}{\partial x} \left[u \frac{\partial w^n}{\partial x} + v \frac{\partial w^n}{\partial y} \right] + \frac{\Delta t}{2} v \frac{\partial}{\partial y} \left[u \frac{\partial w^n}{\partial x} + v \frac{\partial w^n}{\partial y} \right] \end{aligned} \quad (3.94)$$

Os dois últimos termos da equação acima são conhecidos como difusão artificial ou difusão numérica e são eles que atuam para a redução das oscilações espúrias que aparecem para Reynolds moderados ou elevados. Outros esquemas são conhecidos na literatura para eliminar essas oscilações espúrias tais como *Petrov-Galerkin* para equações em 1D e *Streamline Upwind Petrov-Galerkin* (SUPG) para equações em 2D ambos em problemas permanentes. Nesses esquemas, as funções bases são modificadas para obter um efeito *upwind*. Para problemas transientes, além do *Taylor-Galerkin* temos: *Semi-Lagrangiano* e *Galerkin Característico*. Os esquemas *Taylor-Galerkin* e *Galerkin Característico* possuem o mesmo resultado quando a variável é escalar como apresentado por Lohner, Morgan e Zienkiewicz (1984) [38].

Na forma vetorial, as equações de governo discretizadas no tempo possuem a forma:

$$\dot{w} + \mathbf{v} \cdot \nabla w = \frac{1}{Re} \nabla^2 w + \frac{\Delta t}{2} \mathbf{v} \cdot \nabla [\mathbf{v} \cdot \nabla w] \quad (3.95)$$

$$\nabla^2 \psi = -w \quad (3.96)$$

$$\mathbf{v} = \mathbf{D}\psi \quad (3.97)$$

$$\dot{c} + \mathbf{v} \cdot \nabla c = \frac{1}{ReSc} \nabla^2 c + \frac{\Delta t}{2} \mathbf{v} \cdot \nabla [\mathbf{v} \cdot \nabla c] \quad (3.98)$$

onde \dot{w} e \dot{c} são $[w^{n+1} - w^n]/\Delta t$ e $[c^{n+1} - c^n]/\Delta t$ respectivamente, \mathbf{v} é o vetor velocidade cujas componentes são $\mathbf{v} = [u, v]$ e \mathbf{D} é um operador diferencial com componentes $\mathbf{D} = [\partial/\partial y, -\partial/\partial x]$.

3.6 Mesh Elements

A computational mesh can be structured or unstructured and the choice of the element that constitutes this mesh is of vital importance for a good accuracy of the solution. 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* [5] [6]. 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 element*. 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. 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 4:

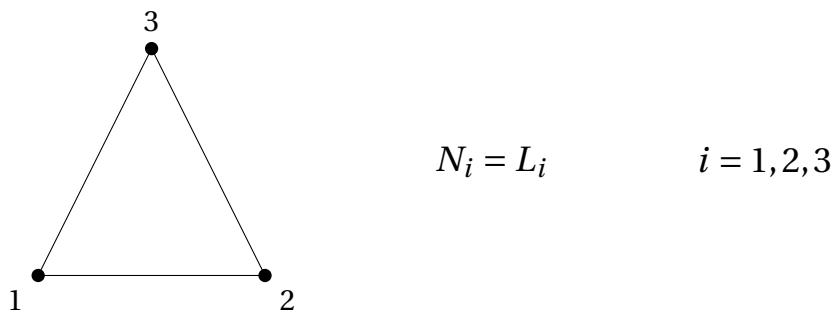


Figure 4: Linear Triangular Element

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 5:

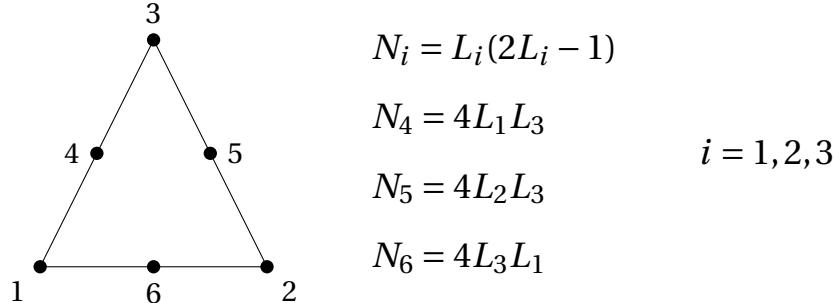


Figure 5: Quad Triangular Element

Mini Triangular Element: This element is 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 as well as quad 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 6:

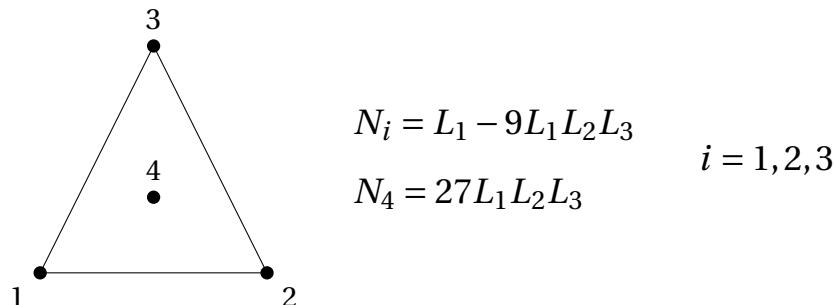


Figure 6: Mini Triangular Element

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 Eqs (3.76 to 3.80) can be represented in matrix form by:

$$\begin{aligned} M\dot{w} + u \cdot G_x w + v \cdot G_y w + \frac{1}{Re} [K_{xx} + K_{yy}] w \\ + \frac{\Delta t}{2} u [u K_{xx} + v K_{yx}] w + \frac{\Delta t}{2} v [u K_{xy} + v K_{yy}] w = 0 \end{aligned} \quad (3.99)$$

$$- [K_{xx} + K_{yy}] \psi + Mw = 0 \quad (3.100)$$

$$Mu - G_y \psi = 0 \quad (3.101)$$

$$Mv + G_x \psi = 0 \quad (3.102)$$

$$\begin{aligned} M\dot{c} + u \cdot G_x c + v \cdot G_y c + \frac{1}{ReSc} [K_{xx} + K_{yy}] c \\ + \frac{\Delta t}{2} u [u K_{xx} + v K_{xy}] c + \frac{\Delta t}{2} v [u K_{yx} + v K_{yy}] c = 0 \end{aligned} \quad (3.103)$$

where the **M**, **G_x**, **G_y**, **K_{xx}**, **K_{xy}**, **K_{yx}**, **K_{yy}** matrices have **np x np** size (where *np* is node number) and they are defined as:

$$\mathbf{M} = \mathbf{A}m^e \quad (3.104)$$

$$\mathbf{G}_x = \mathbf{A}g_x^e \quad (3.105)$$

$$\mathbf{G}_y = \mathbf{A}g_y^e \quad (3.106)$$

$$\mathbf{K}_{xx} = \mathbf{A}k_{xx}^e \quad (3.107)$$

$$\mathbf{K}_{xy} = \mathbf{A}k_{xy}^e \quad (3.108)$$

$$\mathbf{K}_{yx} = \mathbf{A}k_{yx}^e \quad (3.109)$$

$$\mathbf{K}_{yy} = \mathbf{A}k_{yy}^e \quad (3.110)$$

where \mathbf{A} is an assembly operator that it assembles the elementary matrix in global matrix, satisfying the local and global matrix index correspondence and the m^e , g_x^e , g_y^e , k_{xx}^e , k_{xy}^e , k_{yx}^e , k_{yy}^e are elementary matrices whose size for the *linear triangular element* is 3x3 and they are defined by:

$$\begin{aligned} m^e &= \int_{\Omega^e} N_i^e N_j^e d\Omega \\ g_x^e &= \int_{\Omega^e} \frac{\partial N_i^e}{\partial x} N_j^e d\Omega \\ g_y^e &= \int_{\Omega^e} \frac{\partial N_i^e}{\partial y} N_j^e d\Omega \\ k_{xx}^e &= \int_{\Omega^e} \frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} \\ k_{xy}^e &= \int_{\Omega^e} \frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial y} \\ k_{yx}^e &= \int_{\Omega^e} \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial x} \\ k_{yy}^e &= \int_{\Omega^e} \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} \end{aligned} \quad (3.111)$$

Thus, the governing equations in matrix form according to the Finite Element Method

that we used in this work were:

$$\begin{aligned} \frac{M}{\Delta t} w^{n+1} = & \frac{M}{\Delta t} w^n - u \cdot G_x w^n - v \cdot G_y w^n - \frac{1}{Re} [K_{xx} + K_{yy}] w^n \\ & - \frac{\Delta t}{2} u [u K_{xx} + v K_{yx}] w^n - \frac{\Delta t}{2} v [u K_{xy} + v K_{yy}] w^n \end{aligned} \quad (3.112)$$

$$[K_{xx} + K_{yy}] \psi = M w \quad (3.113)$$

$$M u = G_y \psi \quad (3.114)$$

$$M v = -G_x \psi \quad (3.115)$$

$$\begin{aligned} \frac{M}{\Delta t} c^{n+1} = & \frac{M}{\Delta t} c^n - u \cdot G_x c^n - v \cdot G_y c^n - \frac{1}{ReSc} [K_{xx} + K_{yy}] c^n \\ & - \frac{\Delta t}{2} u [u K_{xx} + v K_{yx}] c^n - \frac{\Delta t}{2} v [u K_{xy} + v K_{yy}] c^n \end{aligned} \quad (3.116)$$

4 COMPUTATIONAL CODE

4.1 Introduction

In this chapter, we will present the main characteristics of the computational code developed in Python 2.7 [7] using the object-oriented paradigm (OOP) in order to reuse the code in other simulations in the future. All developed classes are imported into the simulator (*TriSim*), where the result of the numerical simulation is exported as presented in the simplified *Class Diagram* (UML) of Figure 7. Initially, the *script* that performs the import of the computational mesh for the simulation is presented. Then, the assembly of the global matrices is done respecting the correspondence between the global and local index. Later on, we present the application of boundary conditions for both *Dirichlet* and *Neumann*. Finally, the solve algorithm for the vorticity-streamfunction formulation with the species transport equation is presented.

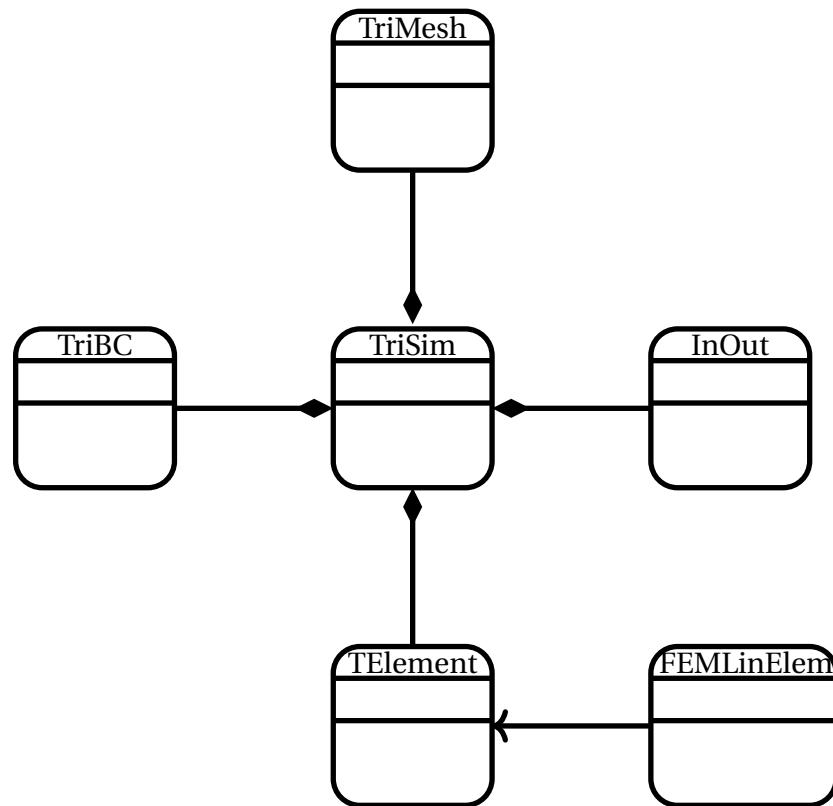


Figure 7: Simplified Class Diagram

4.2 Mesh Import

The domain consists of an unstructured mesh generated by the *GMSH* open-source software proposed by Geuzaine and Remacle (2009) [4] and it was imported into the numerical simulation by the *TriMesh* class. Initially, we used a well-known computational community script to convert the *.msh* file to a *Python* list:

```
-----
mesh = []                                | python list inicialization
with open("file.msh") as mesh:           | open file
    for line in mesh:                   | loop file each line
        row = line.split()              | split file line
        mesh.append(row[:])            | add file line from python list
-----
```

Then, 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)* and *the Boundary nodes*. The Table 1 shows the average processing time for mesh import in several unstructured linear triangular meshes import.

Nodes	Elements	AVG Processing Time (s)
10482	20142	0,6
40819	80005	2,6
249677	495289	16,6
993091	2010501	70,4

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

4.3 Matrix Assembly

After importing the *.msh* file, the global arrays were assembled. They were initialized as sparse matrices by the *Scipy* library [54] and the following *script* was used for the assembly:

```
-----
for e in range(0, ne):                                | element loop
    linear_element(e)                                 | elementary matrix
                                                       by gaussian quadrature

for i in range(0,3):
    ii = IEN[e][i]

for j in range(0,3):
    jj = IEN[e][j]

    Kxx[ii,jj] += kxx_element[i][j]                  |
    Kxy[ii,jj] += kxy_element[i][j]                  |
    Kyx[ii,jj] += kyx_element[i][j]                  |
    Kyy[ii,jj] += kyy_element[i][j]                  | globals matrices assembly
                                                       | with global and local
    Gx[ii,jj] += gx_element[i][j]                  | index correspondence
    Gy[ii,jj] += gy_element[i][j]                  |
                                                       |
    M[ii,jj] += mass_element[i][j]                  |
```

The assembly of the elementary matrices is done by the module *linear_element* whose required parameter is the element number. This module is part of the *TElement* class where it uses the Gaussian Quadrature to calculate the values of the elementary matrices. 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) [55].

Then, the left hand matrix known as *left hand side (LHS)* is created for the streamfunction, velocity and concentration equations respectively:

```
LHS_psi = sps.lil_matrix.copy(K)
LHS_vx = sps.lil_matrix.copy(M)
LHS_vy = sps.lil_matrix.copy(M)
LHS_c = sps.lil_matrix.copy(M)/dt
```

The *LHS* matrix for the vorticity equation is created during the solution algorithm loop to ensure that it will always be initialized using the original global matrices. It is necessary to use the function *copy* because we want to copy the values of the global arrays and not reference them, for more details consult the *Scipy Community* [56]. The Table 2 shows the average processing time for global matrices assembly in several unstructured linear triangular meshes.

Nodes	Elements	AVG Processing Time (s)
10482	20142	72,9
40819	80005	254,3
249677	495289	1664,9
993091	2010501	69059,9

Table 2: Average processing time for global matrices assembly in several unstructured linear triangular meshes

4.4 Boundary Condition Apply

After the global matrices assembly, the boundary conditions are applied. As was said in the section 4.2, during the mesh import it is possible to identify the nodes that they are in the boundary. The condition where the nodes have their values predefined by the problem is known as *Dirichlet condition*. Therefore, these nodes must not be changed as the simulation is performed. In this way, the product between the column of the global matrix whose index is a node with Dirichlet boundary condition and its predefined value as boundary condition

is subtracted from the vector on the right side of the governing equation. Then, we zero the rows and columns of the global matrix that corresponds to the condition index of Dirichlet and we set the value of 1 on the main diagonal.

For example, we will consider a matrix with ($np \times np$) size and node 2 as a node located on the domain boundary where the Dirichlet condition is proposed. Thus, the following algorithm is done as presented by Anjos (2007) [51]:

1. The boundary condition is located in the matrix:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2j} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3j} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

↑ Identify the column whose index is
Dirichlet condition

2. Subtract the product between the column where the matrix node is located and its predefined value with the vector on the right hand side of the equation:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2j} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3j} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

↑ Subtract the product from this column with the
 c_2 value on the right hand side of the equation

that is,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2j} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3j} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix} - \underbrace{\begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \\ \vdots \\ a_{n2} \end{bmatrix}}_{bc_dirichlet} * c_2$$

3. **The column and line of the boundary condition node index in matrix are filled with zeros:**

$$\begin{bmatrix} a_{11} & 0 & a_{13} & \cdots & a_{1j} & \cdots & a_{1n} \\ 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ a_{31} & 0 & a_{33} & \cdots & a_{3j} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a_{n1} & 0 & a_{n3} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix} - \underbrace{\begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \\ \vdots \\ a_{n2} \end{bmatrix}}_{bc_dirichlet} * c_2$$

4. **One is placed on the main diagonal of the matrix whose index is the boundary condition node:**

$$\begin{bmatrix} a_{11} & 0 & a_{13} & \cdots & a_{1j} & \cdots & a_{1n} \\ 0 & 1 & 0 & \cdots & 0 & \cdots & 0 \\ a_{31} & 0 & a_{33} & \cdots & a_{3j} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a_{n1} & 0 & a_{n3} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix} - \underbrace{\begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \\ \vdots \\ a_{n2} \end{bmatrix}}_{bc_dirichlet} * c_2$$

5. **The next node is located and the step is performed again.**

The implementation of this algorithm was performed by the following *script*:

```

for mm in ibc:                                | boundary nodes loop
    bc_dirichlet -= LHS[:,mm]*bc_1[mm]        | step 2
    LHS[:,mm] = 0.0                            | step 3 - zero columns
    LHS[mm,:] = 0.0                            | step 3 - zero rows
    LHS[mm,mm] = 1.0                          | step 4 - one in main diagonal
    bc_dirichlet[mm] = bc_1[mm]                | computing the Dirichlet condition

```

where *ibc* is a list containing all the boundary nodes whose condition is Dirichlet and *bc_1* is an auxiliary vector with *np* size where the value of the Dirichlet condition is computed in each index correspondence. The *-=* symbol ensures that the contribution of each node whose Dirichlet condition is computed. This procedure must be performed for each of the *LHS*. The Table 3 shows the average processing time for the *Dirichlet* condition apply in several unstructured linear triangular meshes.

Nodes	Elements	AVG Processing Time (s)
10482	20142	6,8
40819	80005	37,5
249677	495289	467,7
993091	2010501	3720,6

Table 3: Average processing time for Dirichlet condition in several unstructured linear triangular meshes

Another type of very common boundary condition is one there is a flux in the domain boundaries. This boundary condition is known as *Neumann condition* and in variational formulation it is called *Natural condition*. Unlike the Dirichlet condition, this type of boundary condition does not affect the *LHS* matrix when the flux is constant. We should only add their contribution to the vector on the right hand side of the equation, that is:

$$\begin{bmatrix}
 a_{11} & 0 & a_{13} & \cdots & a_{1j} & \cdots & a_{1n} \\
 0 & 1 & 0 & \cdots & 0 & \cdots & 0 \\
 a_{31} & 0 & a_{33} & \cdots & a_{3j} & \cdots & a_{3n} \\
 \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 a_{n1} & 0 & a_{n3} & \cdots & a_{nj} & \cdots & a_{nn}
 \end{bmatrix}
 \begin{bmatrix}
 c_1 \\
 c_2 \\
 c_3 \\
 \vdots \\
 c_n
 \end{bmatrix}
 =
 \begin{bmatrix}
 b_1 \\
 b_2 \\
 b_3 \\
 \vdots \\
 b_n
 \end{bmatrix}
 -
 \begin{bmatrix}
 a_{12} \\
 a_{22} \\
 a_{32} \\
 \vdots \\
 a_{n2}
 \end{bmatrix}
 * c_2
 +
 \begin{bmatrix}
 ne_1 \\
 ne_2 \\
 ne_3 \\
 \vdots \\
 ne_n
 \end{bmatrix}$$

bc_dirichlet
 bc_neumann

As mentioned in the chapter 3, we have only Dirichlet condition in this work. Below, however, we will present the implementation of this type of condition. In order to exemplify, we will consider the boundary term of the equation 3.24, that is:

$$\frac{1}{ReSc} \int_{\Gamma} \eta \nabla c \cdot \mathbf{n} d\Gamma \tag{4.1}$$

where ∇c is the flux that will be considered constant. After discretization by Galerkin formulation, we have the following expression:

$$\frac{1}{ReSc} \int_{\Gamma} N_j \nabla c \cdot \mathbf{n} d\Gamma \tag{4.2}$$

that is:

$$\frac{1}{ReSc} \left[\frac{\text{length} \nabla c}{2} \right] \tag{4.3}$$

where the *length* is the edge length of element. Whereas a two-dimensional domain, where i is a boundary node, $i - 1$ and $i + 1$ are its neighbors nodes, the edges of node i can be shown as:

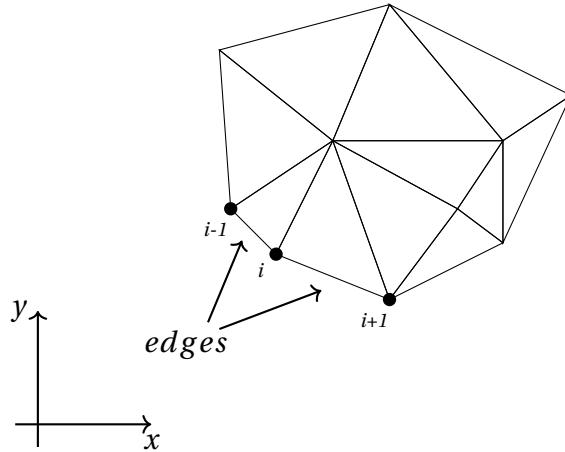


Figure 8: Edges of a node

where the left edge of the i node is composed of $i - 1$ and i , while the right edge is composed of i and $i + 1$. Thereby, we can observe that the node i receives contribution from its neighbors edges. The following *script* is used to implement the Neumann boundary condition when required:

```

for i in range(0, len(neumann_edges)):    | boundary nodes loop

    p1 = neumann_edges[i][1]                | nodes of the edges
    p2 = neumann_edges[i][2]                |

    x = x[p1] - x[p2]                     |
    y = y[p1] - y[p2]                     | edge length calculation
    length = numpy.sqrt(x**2 + y**2)       |

    bc_neumann[p1] += (length*nabla_c) / 2. | computing the Neumann
    bc_neumann[p2] += (length*nabla_c) / 2. | condition

```

where *neumann_edges* is a list containing the nodes presents on an edge whose condition is Neumann, *p1* and *p2* are the nodes presents on the edge, *x* and *y* are the coordinates of each node, *length* is the edge length, *nabla_c* is the dimensionless flux modeled on the physical

problem and *numpy* is a numerical library of *Python* in which we are using the square root function (*numpy.sqrt*). The $+=$ symbol ensures that the contribution of the left and right edges is computed.

4.5 Solve Algorithm

The main difficulty in implementation of vorticity-streamfunction formulation is the boundary conditions of the vorticity. Briefly, the solve algorithm used was:

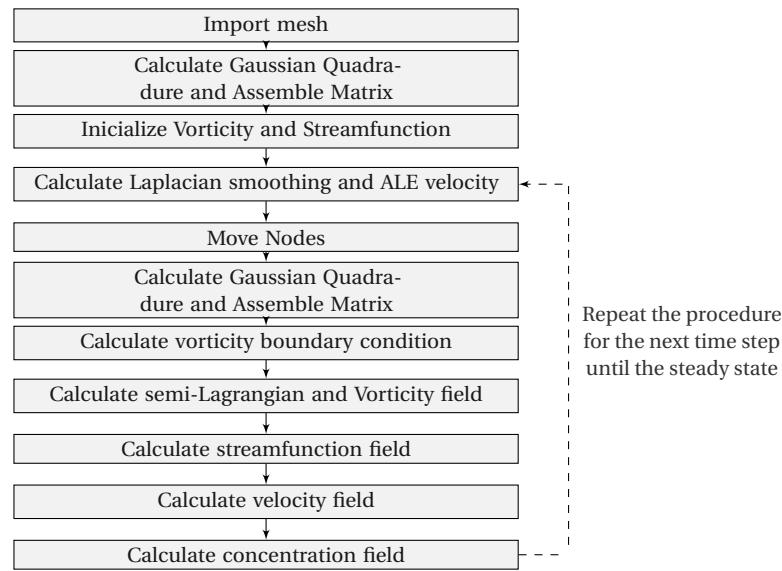


Figure 9: Solve algorithm for Vorticity-Streamfunction Formulation with Species Transport Equation

In order to facilitate its implementation in other research, we will describe each step of the algorithm in detail. The equations are in their matrix form:

1. Initialize Vorticity field:

$$Mw = G_x v - G_y u$$

2. Initialize Streamfunction field:

$$[K_{xx} + K_{yy}] \psi = Mw$$

It is necessary to apply the boundary conditions of ψ in the LHS matrix, its lines and

columns must be zeros and the contribution of the cboundary indices in the vector to the right of the equation, as explained in the section reftriccond.

3. Calcular as condições de contorno da vorticidade utilizando a equação:

$$Mw = G_xv - G_yu$$

Após resolvemos esta equação, possuímos os valores de w para todos os nós do domínio, mas usaremos apenas os nós do contorno para zerar as linhas e as colunas da matriz da equação da vorticidade e sua contribuição no lado direito. Deve-se garantir que a matriz LHS seja inicializada em sua forma original em cada passo de tempo. Para isso, é feito a cada passo de tempo:

```
LHS_w = sps.lil_matrix.copy(M)/dt
```

O script para zerar as linhas e as colunas é semelhante à aplicação da condição de Dirichlet que foi explicado na seção 4.4, exceto à utilização do vetor auxiliar bc_1 que foi substituído pelo w calculado na equação $Mw = G_xv - G_yu$

```
for mm in ibc:                      | loop sobre os nós do contorno de w
    bc_dirichlet = LHS[:,mm]*w[mm]   | o vetor bc_1 é substituído por w
    LHS[:,mm] = 0.0
    LHS[mm,:] = 0.0
    LHS[mm,mm] = 1.0
    bc_dirichlet[mm] = w[mm]         | o vetor bc_1 é substituído por w
```

4. Calcular a vorticidade pela equação:

$$\begin{aligned} \frac{M}{\Delta t}w^{n+1} &= \frac{M}{\Delta t}w^n - u \cdot G_x w^n - v \cdot G_y w^n - \frac{1}{Re} [K_{xx} + K_{yy}] w^n \\ &\quad - u \frac{\Delta t}{2} [u K_{xx} + v K_{yx}] w^n - v \frac{\Delta t}{2} [u K_{xy} + v K_{yy}] w^n \end{aligned}$$

onde w^n é a vorticidade calculada no passo de tempo anterior e w^{n+1} é a vorticidade que será calculada no passo de tempo em análise. É necessário aplicar as condições de contorno de w na matriz a esquerda da equação zerando suas linhas e colunas e a contribuição das colunas dos índices de contorno no vetor a direita da equação, como foi explicado no passo 3.

5. Calcular a função de corrente pela equação:

$$\left[K_{xx} + K_{yy} \right] \psi = Mw$$

É necessário aplicar as condições de contorno de ψ na matriz a esquerda da equação zerando suas linhas e colunas e a contribuição das colunas dos índices de contorno no vetor a direita da equação, conforme explicado na seção 4.4.

6. Calcular a velocidade pela equação:

$$Mu = G_y \psi$$

$$Mv = -G_x \psi$$

É necessário aplicar as condições de contorno de u e v em suas respectivas matrizes a esquerda de cada equação zerando suas linhas e colunas e a contribuição das colunas dos índices de contorno no vetor a direita de cada equação, conforme explicado na seção anterior.

7. Calcular a concentração pela equação:

$$\begin{aligned} \frac{M}{\Delta t} c^{n+1} &= \frac{M}{\Delta t} c^n - u \cdot G_x c^n - v \cdot G_y c^n - \frac{1}{ReSc} \left[K_{xx} + K_{yy} \right] c^n \\ &\quad - u \frac{\Delta t}{2} [u K_{xx} + v K_{yx}] c^n - v \frac{\Delta t}{2} [u K_{xy} + v K_{yy}] c^n \end{aligned}$$

Onde c^n é a concentração calculada no passo de tempo anterior e c^{n+1} é a concentração que será calculada no passo de tempo em análise. É necessário aplicar as condições de contorno de c na matriz a esquerda da equação zerando suas linhas e colunas e a contribuição das colunas dos índices de contorno no vetor a direita da equação, conforme explicado na seção anterior.

8. Retornar ao passo 3 e repetir o procedimento para outro passo de tempo.

Os passos 1 e 2 ficam fora do *loop* do tempo, enquanto os passos de 3 a 7 encontram-se dentro do *loop*. O procedimento para zerar as linhas e as colunas das matrizes globais pode ser feito antes do *loop*, exceto para o caso da vorticidade em que a cada passo de tempo devemos zerar as linhas e as colunas das matrizes globais e atribuir a contribuição dessas colunas no vetor a direita da equação.

A seguir será apresentado o *script* usado na solução da equação da vorticidade. A mesma ideia foi realizada para cada equação de governo, alterando apenas o vetor do lado direito e as contribuições das condições de contorno. Foi utilizado um método iterativo de solução para sistemas lineares conhecido como *Gradientes Conjugados*.

```
RHS = sps.lil_matrix.dot(np.copy(M)/dt,w) \
    - np.multiply(vx,sps.lil_matrix.dot(Gx,w)) \
    - np.multiply(vy,sps.lil_matrix.dot(Gy,w)) \
    - (1.0/Re)*sps.lil_matrix((Kxx+Kyy),w) \
    - (dt/2.0)*np.multiply(u,(np.multiply(u,sps.lil_matrix.dot(Kxx,w))) \
        + np.multiply(v,sps.lil_matrix.dot(Kyx,w)))) \
    - (dt/2.0)*np.multiply(v,(np.multiply(u,sps.lil_matrix.dot(Kxy,w))) \
        + np.multiply(v,sps.lil_matrix.dot(Kyy,w)))) \
RHS = RHS + (1/Re)*bc_neumann
RHS = np.multiply(RHS,bc_2)
RHS = RHS + bc_dirichlet

w = scipy.sparse.linalg.cg(LHS,RHS,w, maxiter=1.0e+05, tol=1.0e-05)
w = w[0].reshape((len(w[0]),1))
```

Onde *RHS* é o vetor do lado direito da equação e significa *right hand side* e *bc_2* é um vetor auxiliar no qual garante que somente os nós sem condição de Dirichlet sejam resolvidos. Ele consiste em um vetor com dimensões *np* onde possui o valor de 1 nos índices cujos nós não possuem condição de Dirichlet e o valor 0 nos índices restantes, isto é, os nós que possuem condição de Dirichlet. Vale ressaltar que o vetor *bc_2* é diferente para cada equação já que os

contornos cuja condição é de Dirichlet variam de equação para equação, ou seja, o vetor bc_2 da equação da vorticidade é diferente da equação de concentração. O primeiro bloco do *script* acima consiste no lado direito da equação (3.112). O segundo bloco refere-se a contribuição das condições de Neumann (para esta simulação é nula) e de Dirichlet, além da aplicação do vetor auxiliar bc_2 . O terceiro bloco consiste na solução da equação pelo método iterativo dos *Gradientes Conjugados*.

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. 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* 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 flow in a cavity with a moving lid (*lid-driven cavity flow*) and the solution is compared with the results presented by Ghia et al. (1982) [8] and Marchi et al. (2009) [9]. In the fifth section, the comparison of the Galerkin and Taylor-Galerkin Method 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 10 presents schematically this flow and the profile of the expected velocity field.

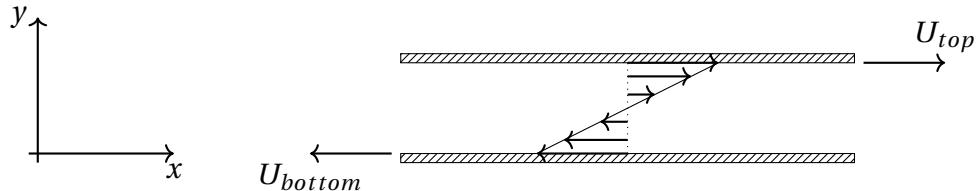


Figure 10: Couette Flow

The velocity profile equation is shown below:

$$u = [U_{top} - U_{bottom}] \frac{y}{L} + U_{bottom} \quad (5.1)$$

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 domain was discretized using a linear triangular mesh with 3835 nodes and 7299 elements.

The Figure 11 shows the unsteady velocity profile when $Re = 100$, in addition to the comparison between the numerical solution and the analytical solution in the steady state of the proposed problem. It is possible to observe that the numerical solution converges to the analytical solution when the flow becomes steady.

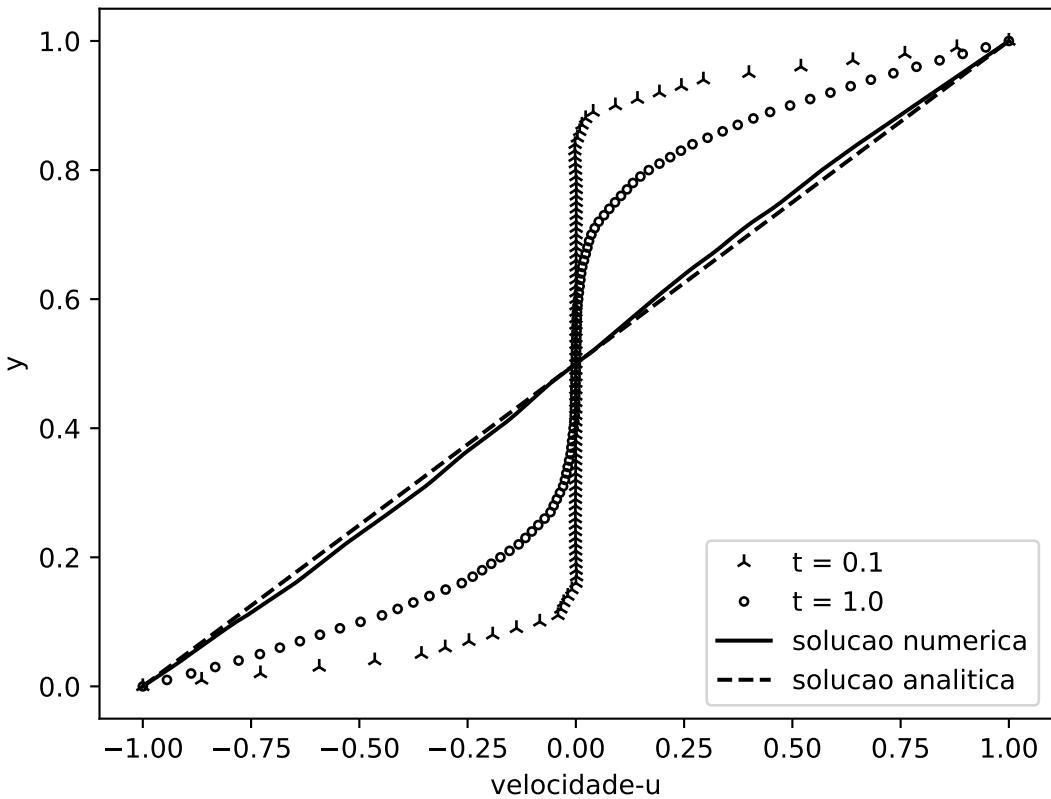


Figure 11: Unsteady velocity profile when $Re = 100$ and the comparison between the numerical and analytical solution for Couette flow.

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) [47]. This flow is known as *Poiseuille flow*. The Figure 12 presents schematically this flow and the expected velocity field.

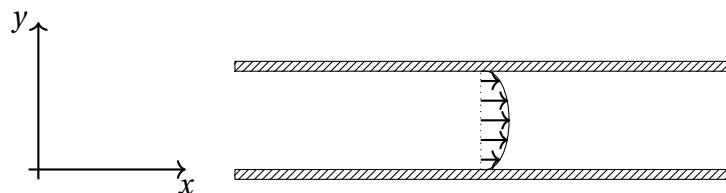


Figure 12: Poiseuille Flow

The velocity profile equation is shown below:

$$u = \frac{4u_{max}}{L^2} y [L - y] \quad (5.2)$$

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 13 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.

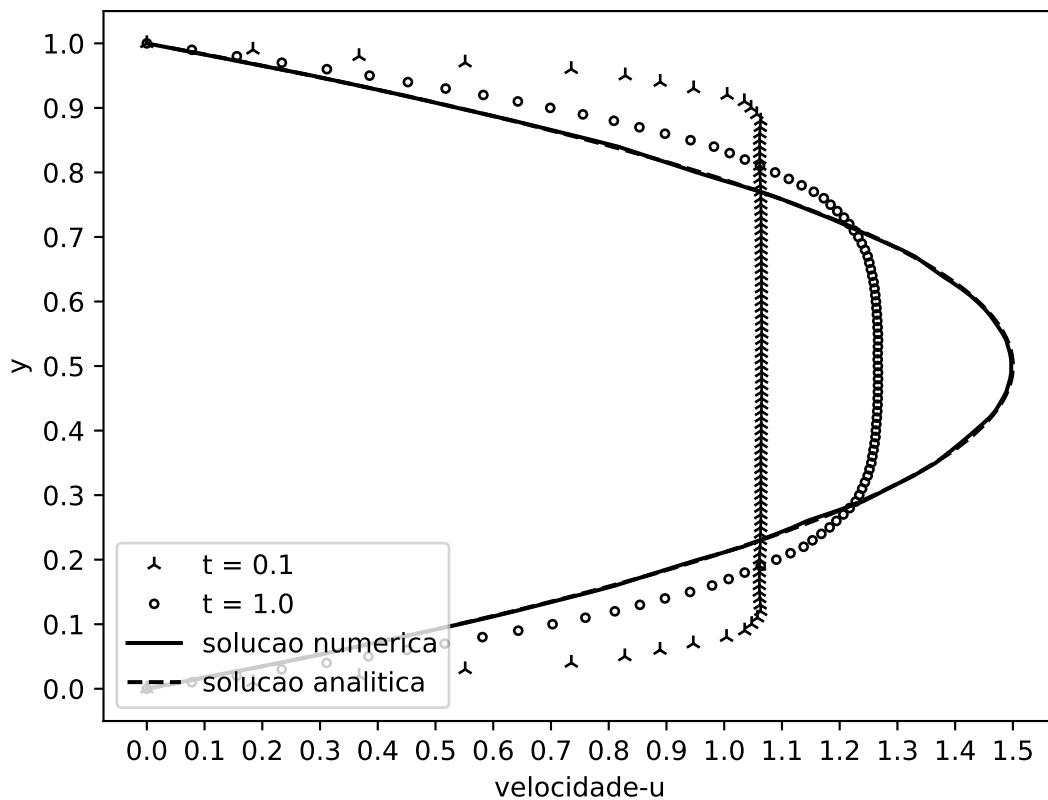


Figure 13: Unsteady velocity profile when $Re = 100$ and the comparison between the numerical and analytical solution for Poiseuille flow.

The Table 4 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 between 0.49% and 0.13%.

Elements	Error (%)
100	25,00
400	7,27
1600	1,94
6400	0,49
25600	0,13

Table 4: The relative error of numerical solution for several elements numbers in an unstructured linear triangular mesh.

The relative error was estimated as:

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

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

The Figure 14 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.

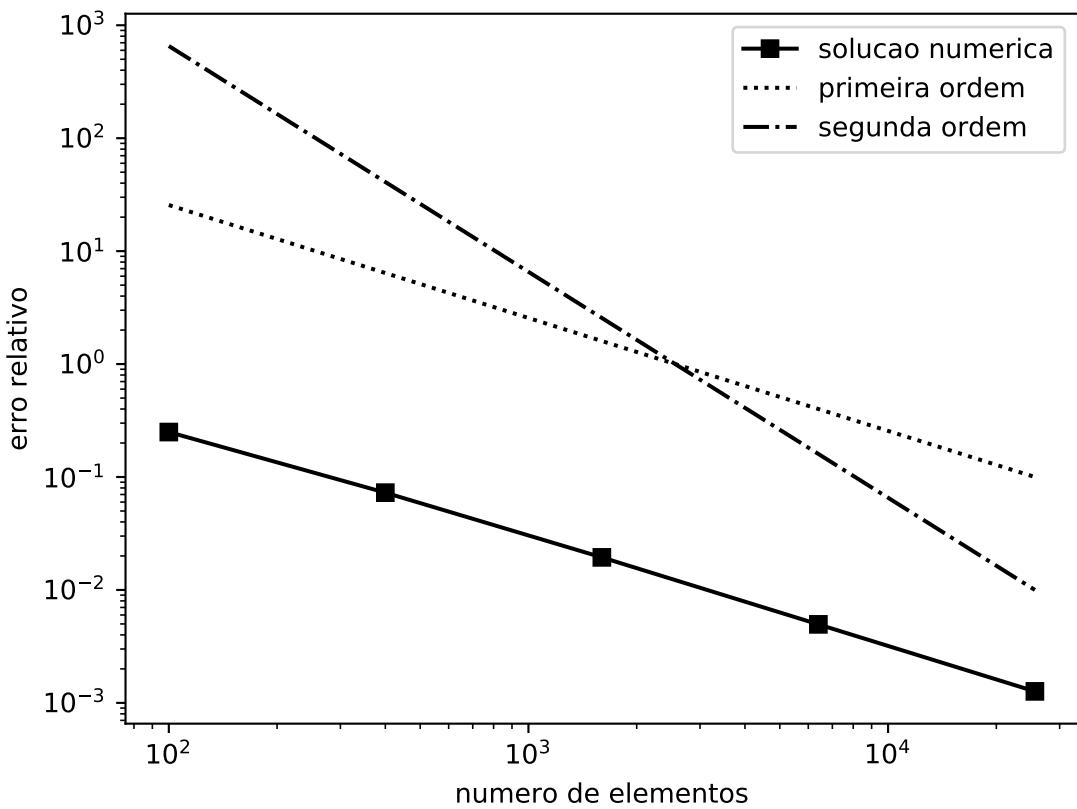


Figure 14: Convergence order in log-log scale: It is estimated the the relative error of numerical solution has first order convergence.

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 15 presents schematically this flow with the specified axis of symmetry and the expected velocity field.

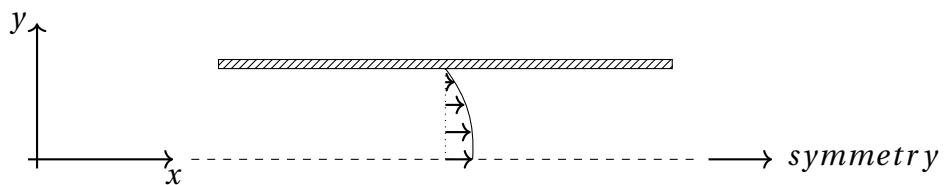


Figure 15: Half Poiseuille flow

The velocity profile equation is shown below:

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

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 16 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.

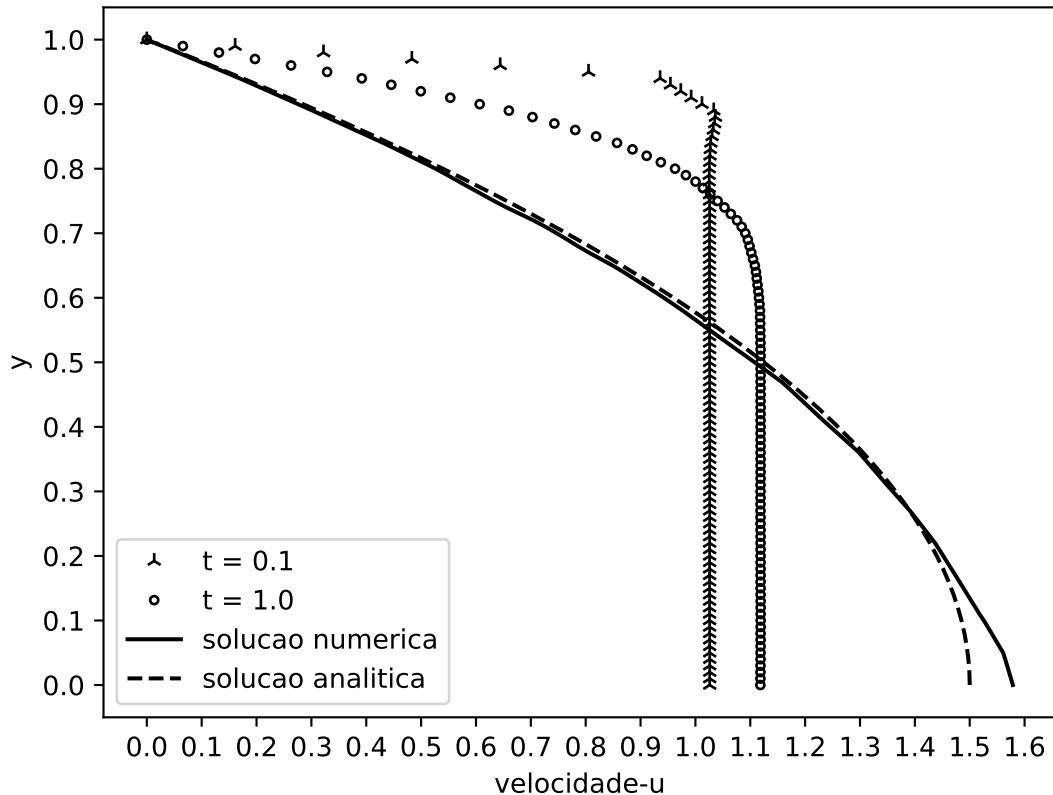


Figure 16: Unsteady velocity profile when $Re = 100$ and the comparison between the numerical and analytical solution for Half Poiseuille flow.

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 17 presents schematically this flow and the expected velocity field.

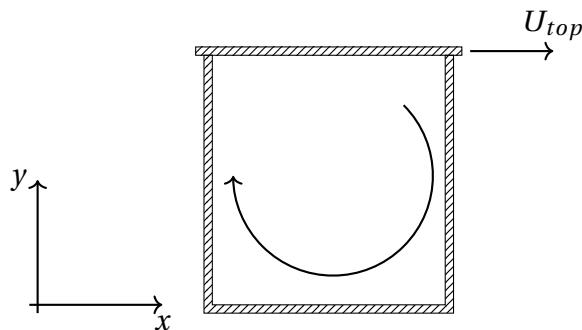


Figure 17: Lid-driven Cavity Flow

The benchmark problem were simulated with the following Reynolds numbers (Re): 10, 100, 400 and 1000. The Figure 18 and Figure 19 prsent the profile of u and v , respectively, for steady state in several Reynolds number. They were compared with Ghia et al. (1982) [8] and Marchi et al. (2009) [9]. The domain was discretized using a linear triangular mesh with 1563 nodes and 2988 elements.

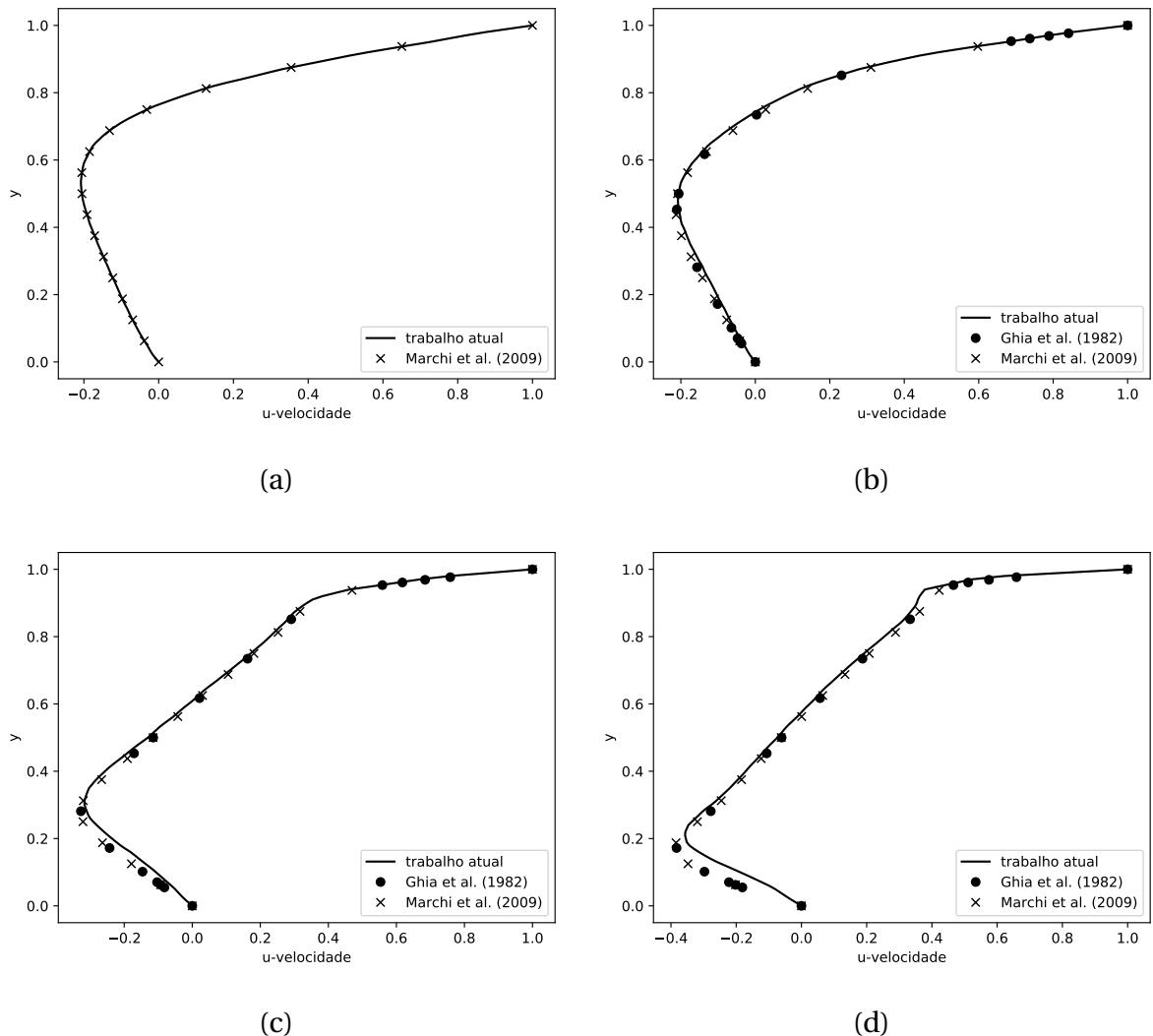


Figure 18: The tangential velocity u profile in central line of cavity ($x = 0.5$) for several Reynolds number: (a) 10 (b) 100 (c) 400 (d) 1000.

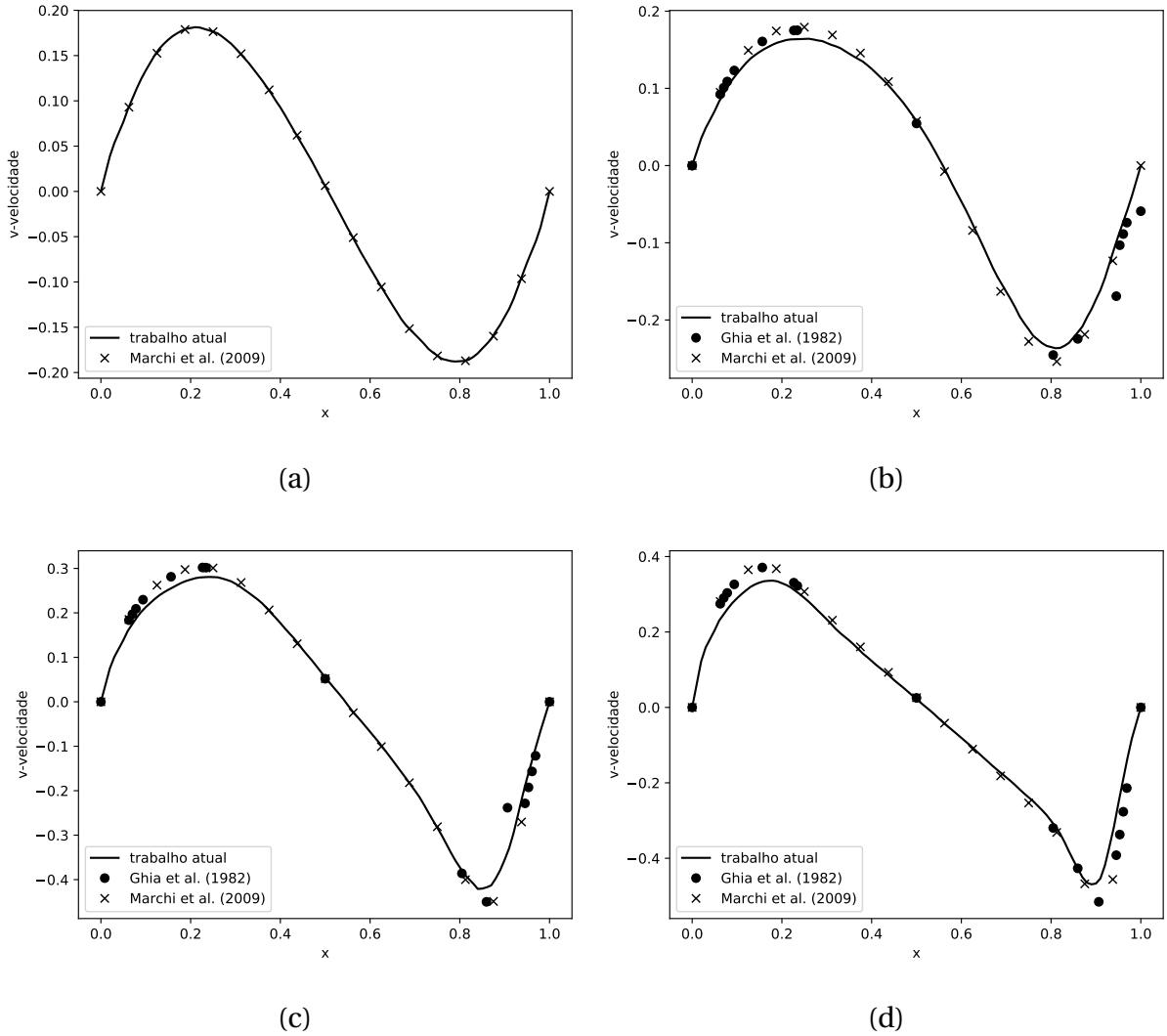


Figure 19: The normal velocity v profile in central line of cavity ($y = 0.5$) for several Reynolds number: (a) 10 (b) 100 (c) 400 (d) 1000.

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 *Taylor-Galerkin* scheme to reduce spurious oscillations compared to the *conventional Galerkin Method*. The Figure 20 presents schematically the problem and the dynamics of scalar transport.

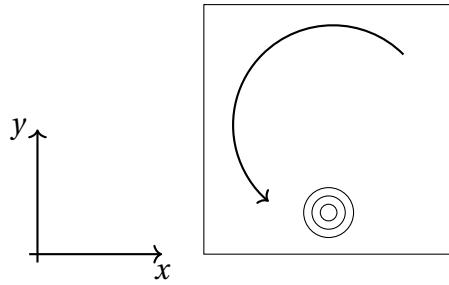


Figure 20: Transport of a scalar in Pure Advection Flow.

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

$$\frac{\partial c}{\partial t} + \mathbf{v} \cdot \nabla c = 0 \quad (5.5)$$

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 a linear triangular mesh with 3835 nodes and 7299 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 21 presents the comparison between the scalar field profile c for the Galerkin and Taylor-Galerkin methods in several positions on the axis of rotation as the flow occurs. It is possible to observe that in both methods the spurious oscillations are presented. In the Taylor-Galerkin method, however, such oscillations are damped in contrast to the Galerkin method where we can observe that spurious oscillations increase and the scalar field profile becomes completely distorted.

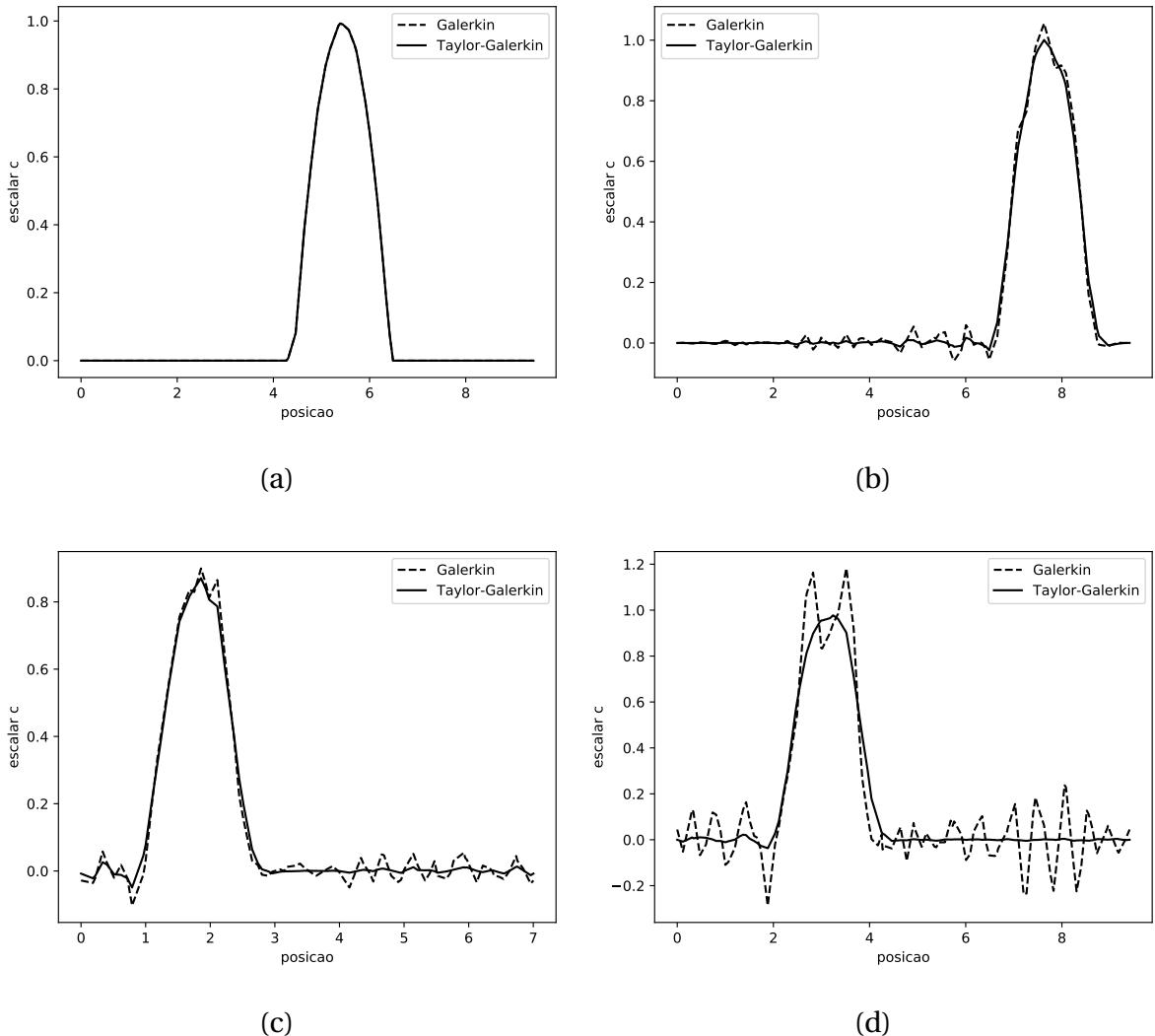


Figure 21: Comparison of c profile for the Galerkin e Taylor-Galerkin method in several positions on the axis of rotation: (a) initial, (b) 1/4 rotation, (c) 1/2 rotation and (d) 3/4 rotation.

The Figure 22 and Figure 23 show the spatial arrangement of spurious oscillations for the Galerkin and Taylor-Galerkin methods respectively. As mentioned earlier, the oscillations presented in the Galerkin method completely distort the scalar field c while in the Taylor-Galerkin method, they are damped as expected. Therefore, for problems where spurious oscillations are present, the Taylor-Galerkin method is superior to the Galerkin method.

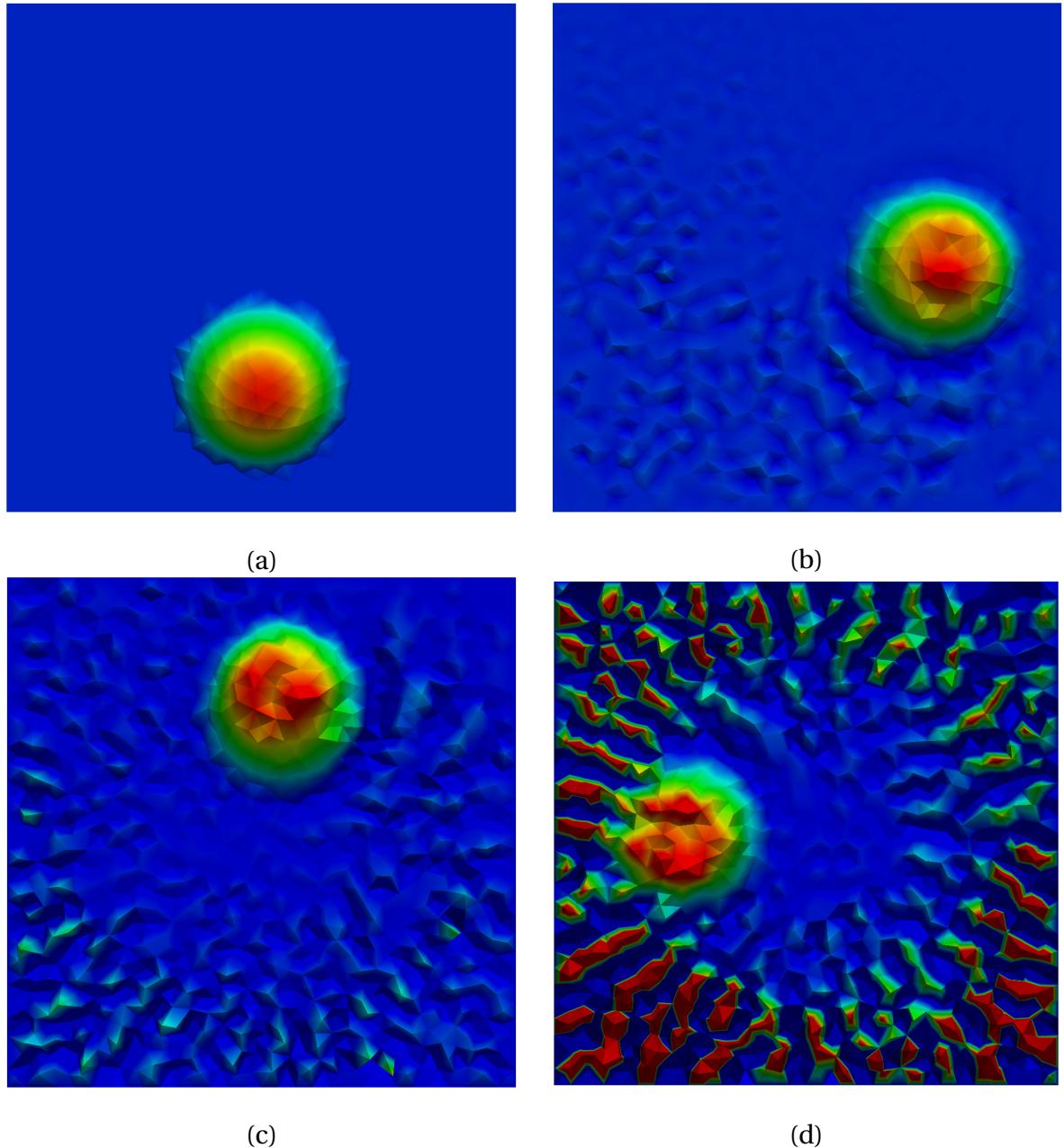


Figure 22: Spurious oscillations for the Galerkin method in several positions of the axis of rotation: (a) initial, (b) 1/4 rotation, (c) 1/2 rotation and (d) 3/4 rotation.

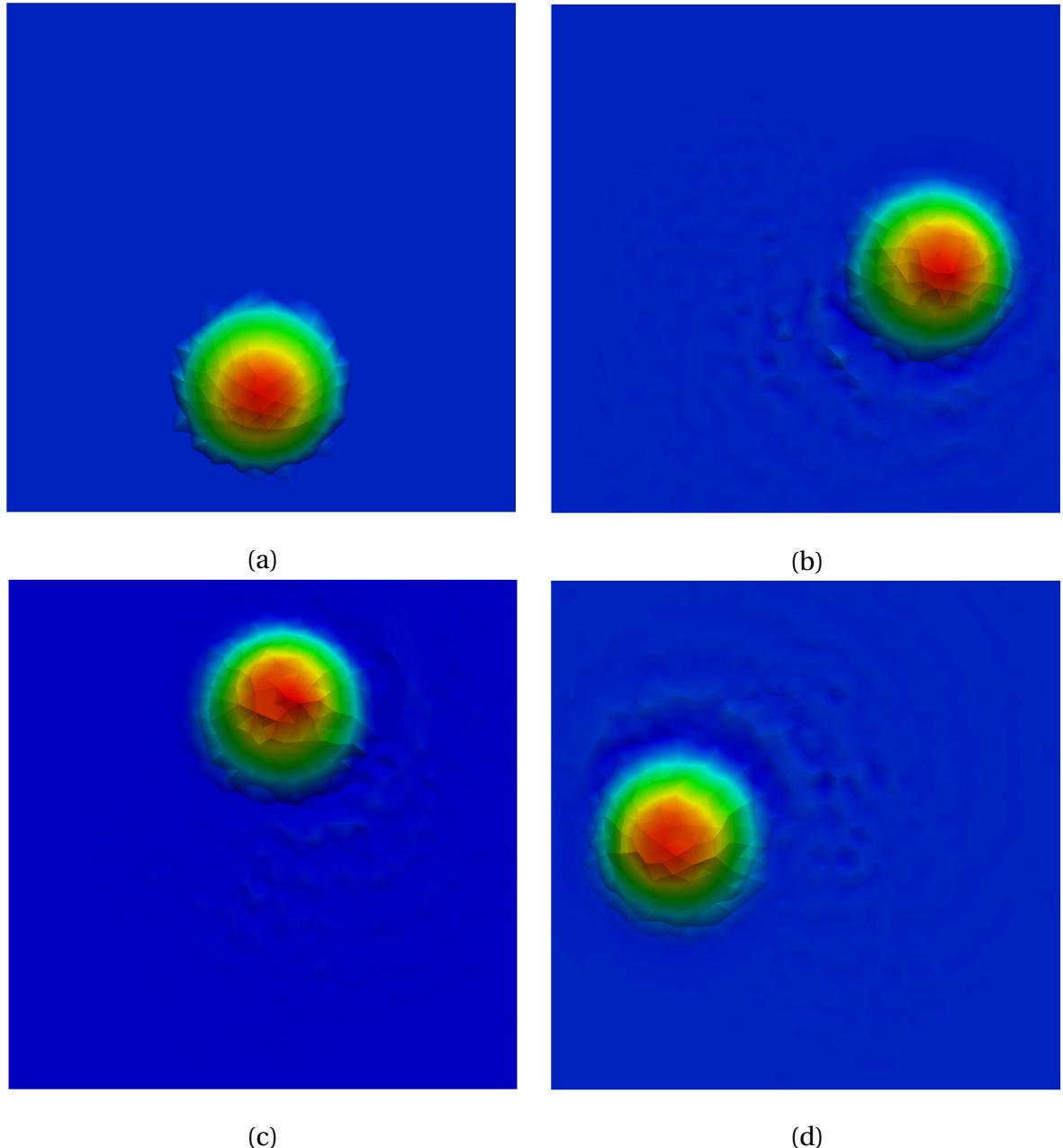


Figure 23: Spurious oscillations for the Taylor-Galerkin method in several positions of the axis of rotation: (a) initial, (b) 1/4 rotation, (c) 1/2 rotation and (d) 3/4 rotation.

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) [19]. According to Kessler et al. (1998) [57], 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 four geometries proposed by Wang et al. (2017) [10], however modified to cartesian coordinates as shown in Figure 24. In the 6.2 section, the coronary artery with atherosclerosis is modeled as a flow in a curved channel. In the section 6.3, the numerical simulation for the coronary artery with atherosclerosis and a drug-eluting stent is presented for several *Schmidt* number, such as $Sc = 1$ and 10 . In the 6.4 section, a numerical simulation of a real coronary artery with atherosclerosis is presented and in the 6.5, the real coronary artery with atherosclerosis and a drug-eluting stent is simulated with several numbers of *Schmidt* number as in the case of the section 6.3. Due to symmetry, only half of the domain was simulated. The simulation was visualized using the *Paraview* open-source software proposed by Henderson (2007) [11].

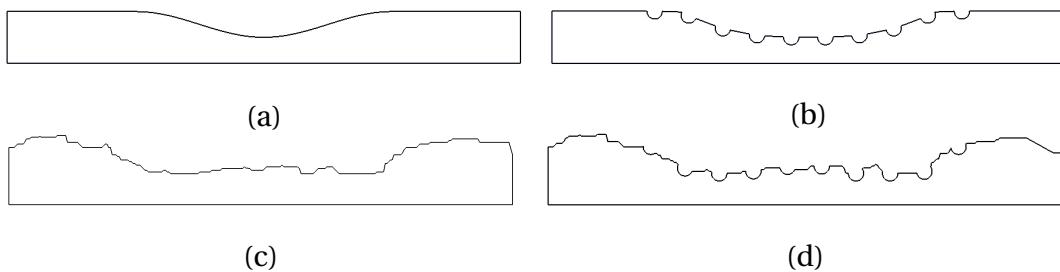


Figure 24: Non-dimensional Domaian for blood flow in coronary artery. The radius used was $R = 1$ and the lumen length was $L = 10R$. (a) Curved Channel (b) Curved Channel with Drug-Eluting Stent (c) Real Channel (d) Real Channel with Drug-Eluting Stent.

6.2 Curved Channel

For the case where the coronary artery has atherosclerosis, the problem is modeled as a flow between curved plates. 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 10261 nodes and 23049 linear triangular elements.

The Figure 25 shows the unsteady velocity profile in the middle of the channel ($x = 5R$). As we can see, the maximum non-dimensional value of the velocity field reaches $u = 2.3$ when the artery has atherosclerosis, that is, there is an increase of 53% of the maximum velocity when compared to the artery without atherosclerosis as shown in Figure 16.

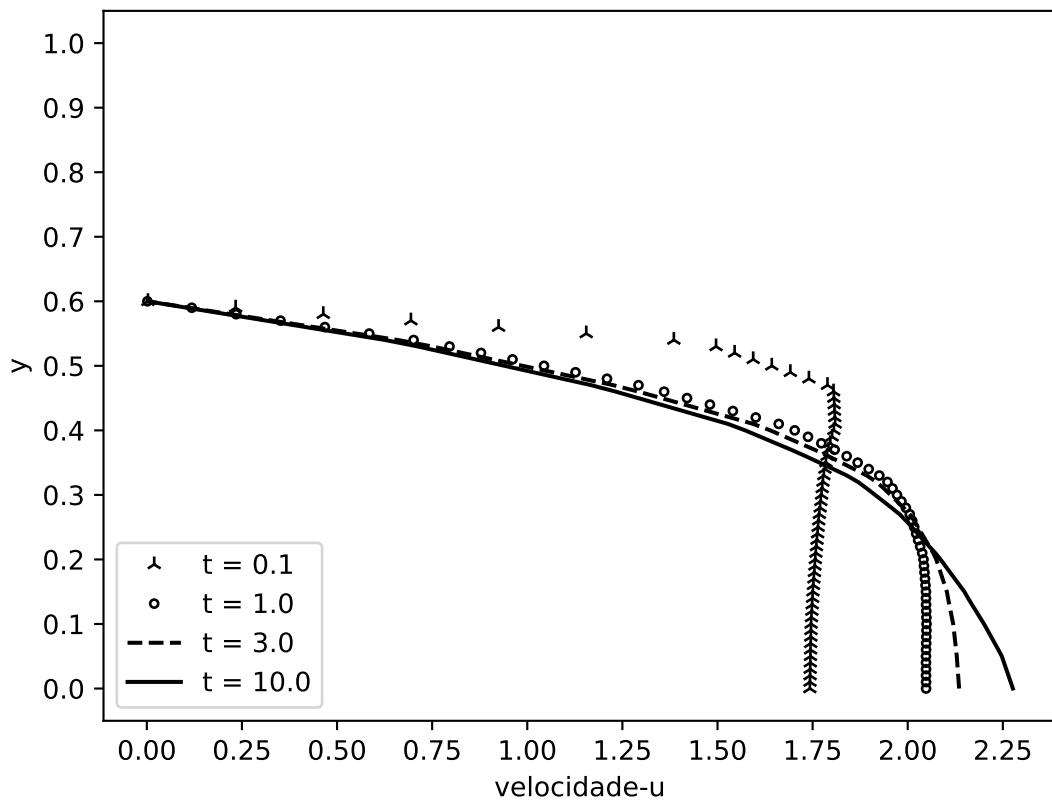


Figure 25: The unsteady velocity profile for the curved channel.

The Figure 26 shows 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 value $u = 2.3$ and the blue color $u = 0$ approximately. Converting to dimensional values we have $u = 27.6\text{cm/s}$ and $u = 0\text{cm/s}$ respectively.

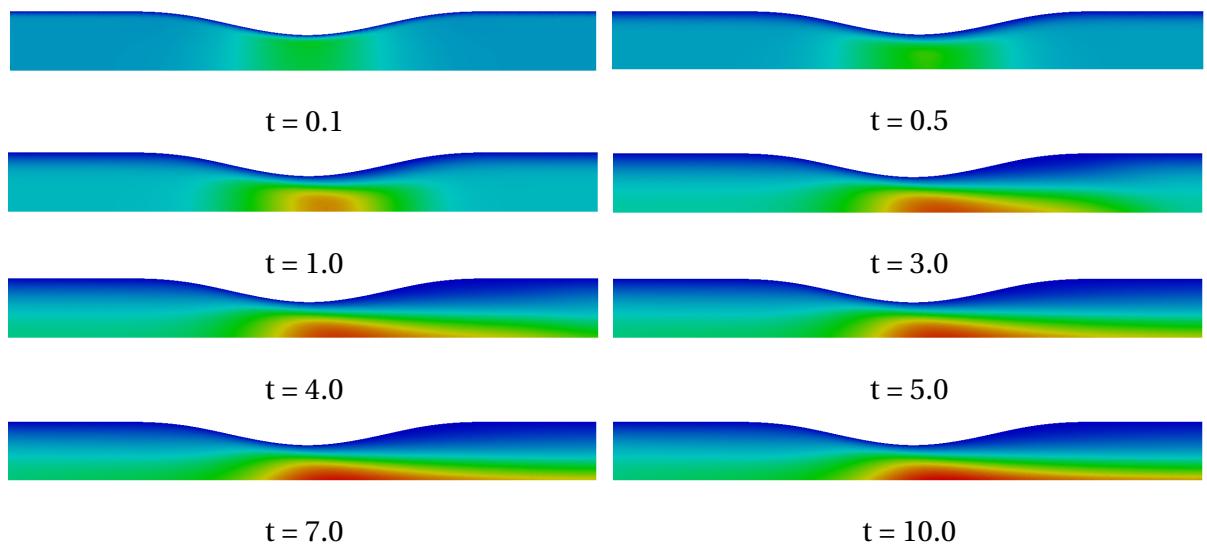


Figure 26: Time and space evolution of the velocity field for curved channel.

6.3 Curved Channel with Drug-Eluting Stent

For this case, the drug-eluting stent is placed on top of the curved channel. It is modeled by 10 uniformly spaced semi-circles. As in the previous case, an channel obstruction of 40% was considered due to atherosclerosis and the domain was discretized using 15875 nodes and 35408 linear triangular elements.

The Figure 27 shows the unsteady velocity profile in the middle of the channel ($x = 5R$). As we can see, the maximum non-dimensional value of the velocity field reaches $u = 3.6$ when the stent is placed, that is, we have an increase of 56% when compared to the artery with only atherosclerosis as in the previous case (see section 6.2).

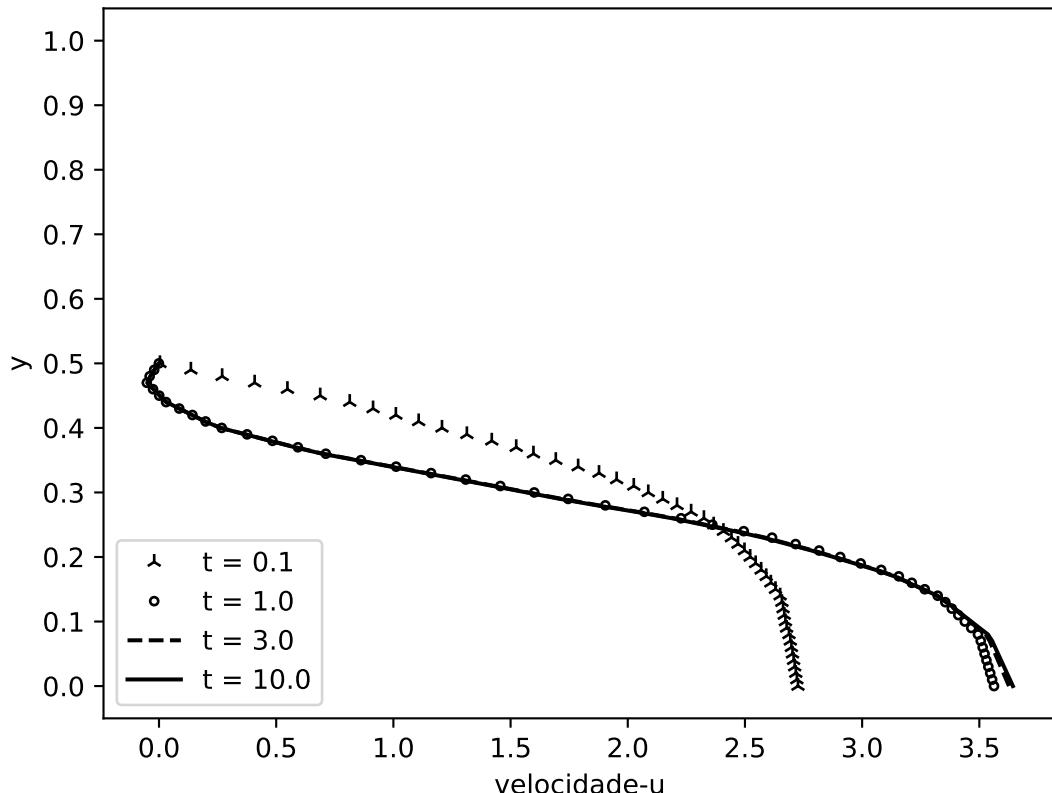


Figure 27: The unsteady velocity profile for curved channel with drug-eluting stent.

The Figure 28 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 $u = 3.6$ value and the blue color $u = 0$ value. Converting to dimensional values, we have $u = 43.2\text{cm/s}$ and $u = 0\text{cm/s}$ respectively.

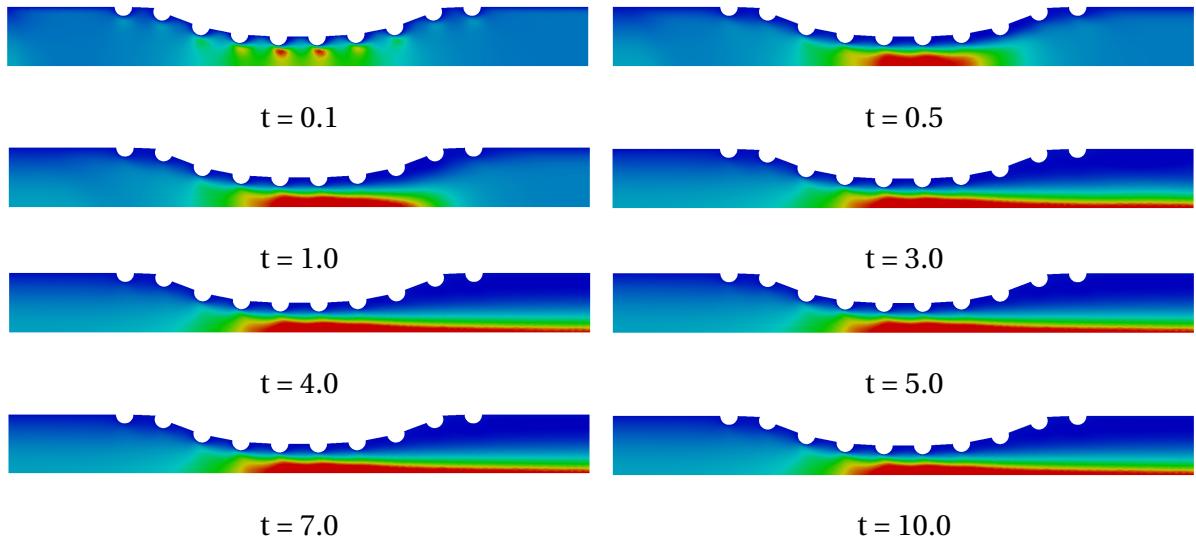


Figure 28: Time and space evolution of the velocity field for curved channel with drug-eluting stent.

As mentioned by Lucena et al. (2017) [58], it is estimated that 47% of the drug is diffused to the lumem and it is lost to the bloodstream. The Figure 29 and Figure 30 show the time and space evolution of the concentration field for several *Schmidt* number, such as: 1 and 10 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 *Schmidt* number directly influences the drug transport in the blood flow. For high values of the *Schmidt* number, the transport of chemical species becomes purely convective.

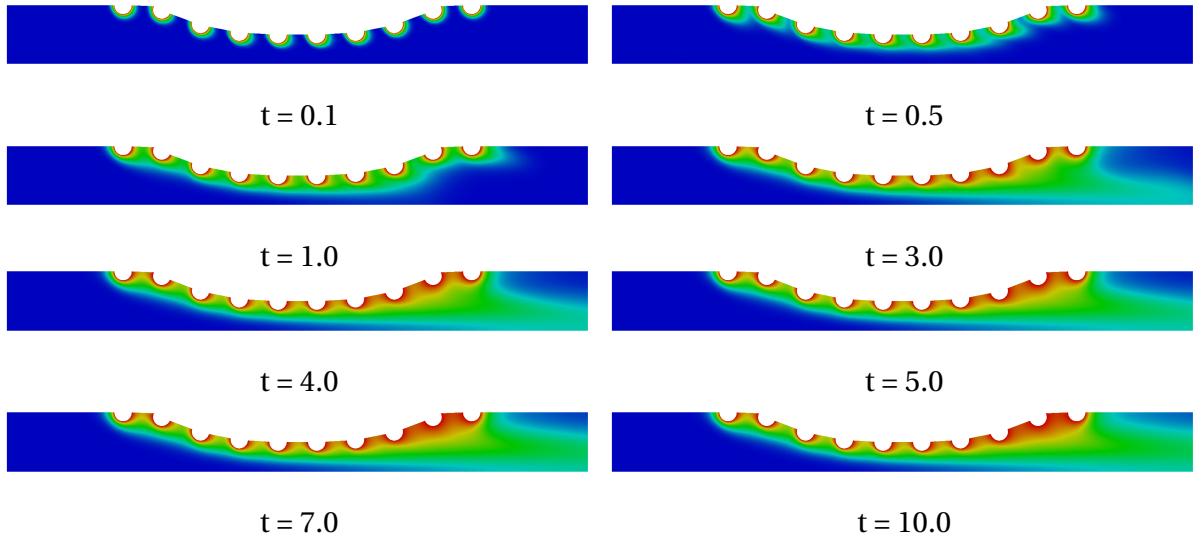


Figure 29: Time and space evolution of the concentration field for curved channel with drug-eluting stent with $Sc = 1$.

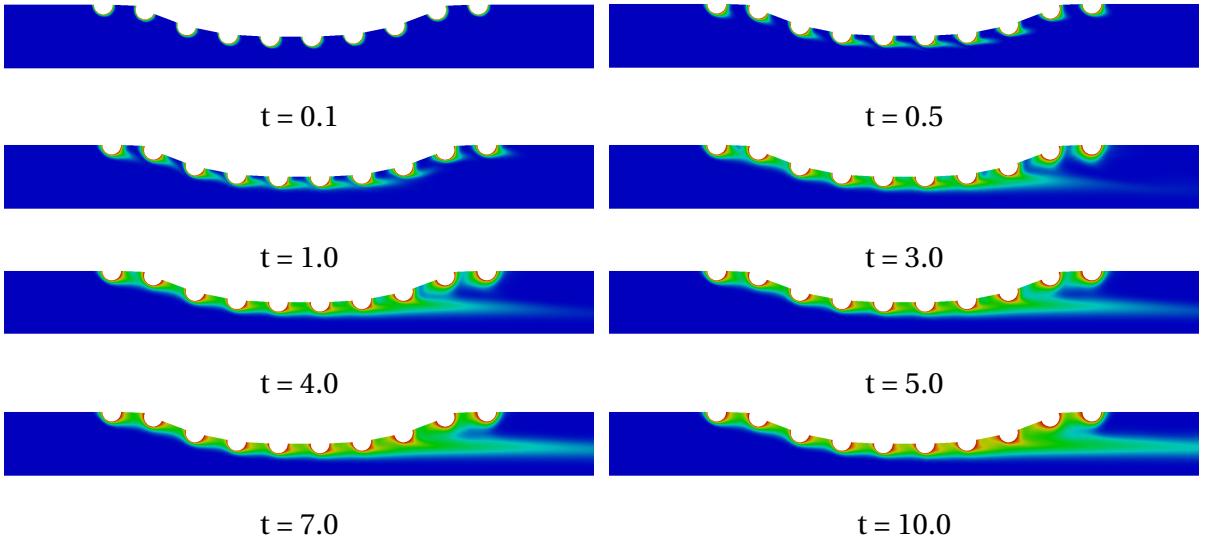


Figure 30: Time and space evolution of the concentration field for curved channel with drug-eluting stent with $Sc = 10$.

6.4 Real Channel

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) [10]. This geometry is particular to each patient due to the patient health conditions. As in the previous cases, an channel obstruction of 40% was considered due to

atherosclerosis and the domain was discretized using 7632 nodes and 14665 linear triangular elements.

The Figure 31 shows the velocity profile in the middle of the channel ($x = 5R$). The maximum non-dimensional value of the velocity field reaches $u = 2.25$. Thus, the curved geometry represents a good approximation as seen in Figure 25, such that the difference of maximum non-dimensional velocity value is about 2%.. .

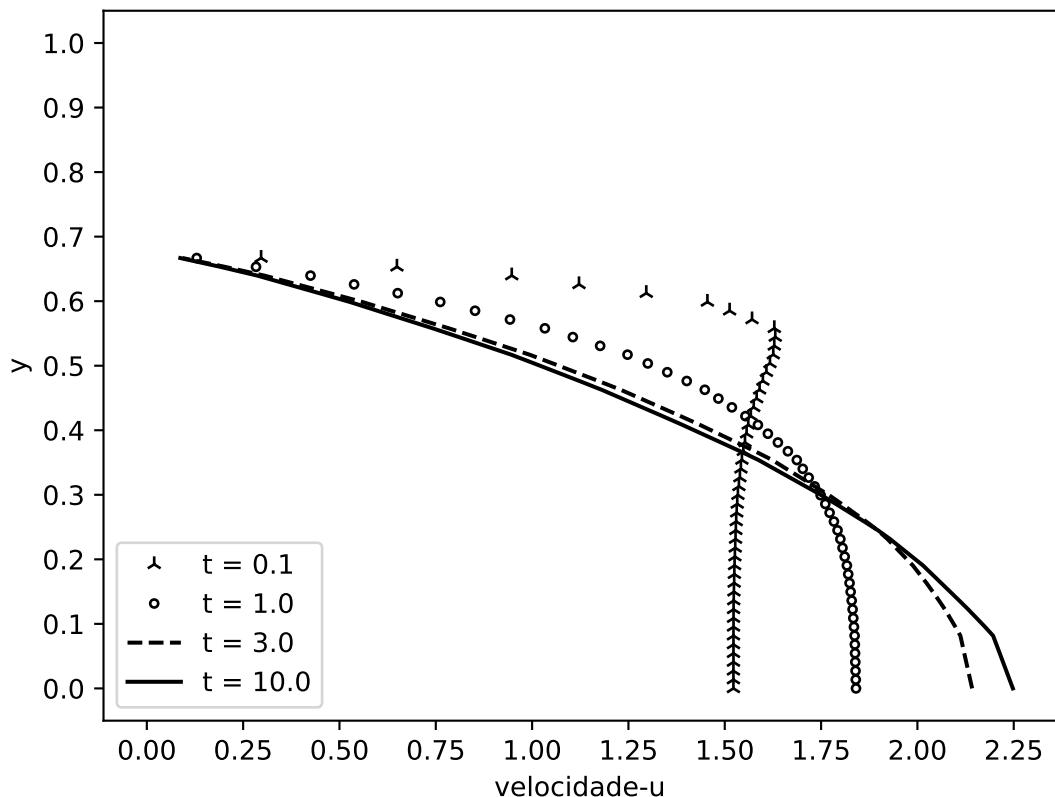


Figure 31: The unsteady velocity profile for the real channel.

The Figure 32 shows 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 value $u = 2.25$ and the blue color $u = 0$ approximately. Converting to dimensional values we have $u = 27.0\text{cm/s}$ and $u = 0\text{cm/s}$ respectively.

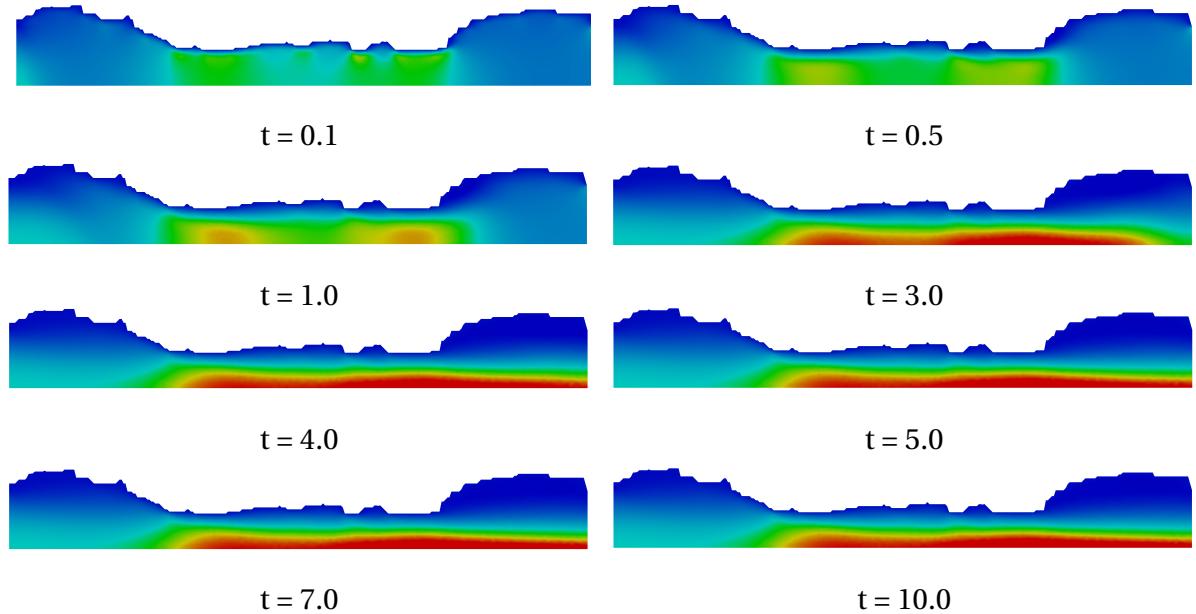


Figure 32: Time and space evolution of the velocity field for real channel.

6.5 Real Channel with Drug-Eluting Stent

For this case, the drug-eluting stent is placed on top of the curved channel. It is modeled by 10 uniformly spaced semi-circles. As in the previous case, an channel obstruction of 40% was considered due to atherosclerosis and the domain was discretized using 11807 nodes and 26426 linear triangular elements.

The Figure 33 shows the unsteady velocity profile in the middle of the channel ($x = 5R$). As we can see, the maximum non-dimensional value of the velocity field reaches $u = 2.65$ when the stent is placed, that is, we have an increase of 18% when compared to the artery with only atherosclerosis as in the previous case (see section 6.4). However, this increase of the velocity may vary according to the coronary artery geometry for each patient.

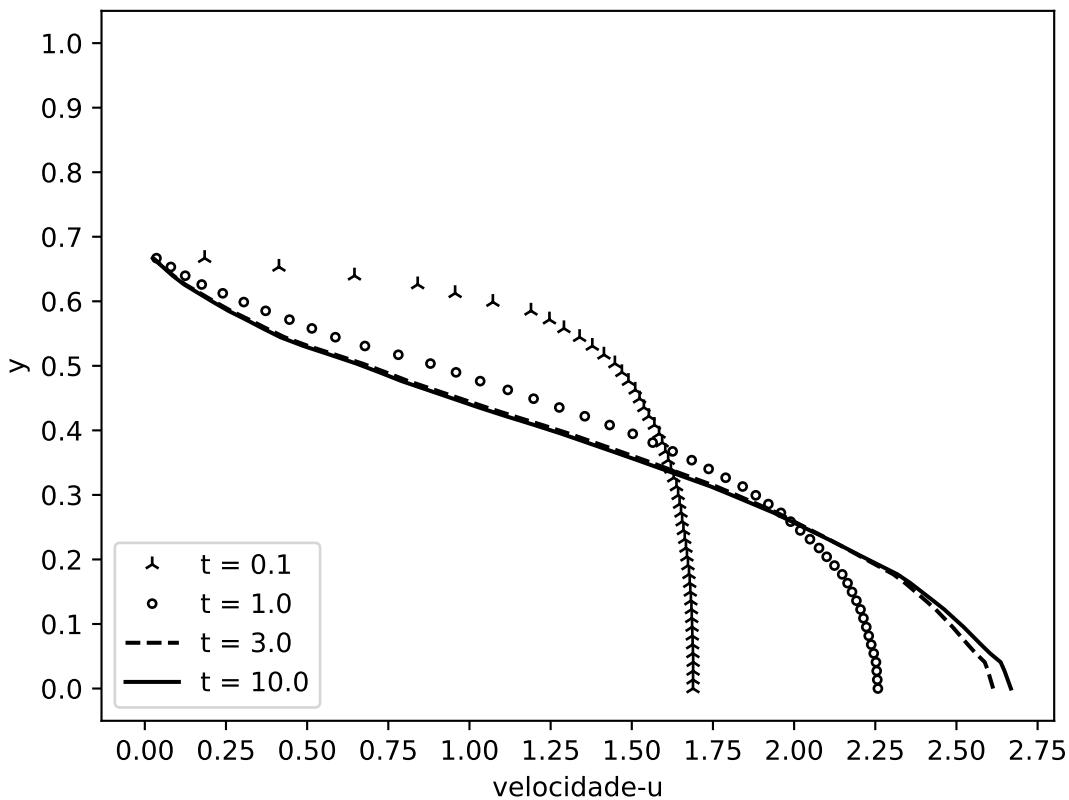


Figure 33: The unsteady velocity profile for real channel with drug-eluting stent.

The Figure 34 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 $u = 2.65$ value and the blue color $u = 0$ value. Converting to dimensional values, we have $u = 31.8 \text{ cm/s}$ and $u = 0 \text{ cm/s}$ respectively.

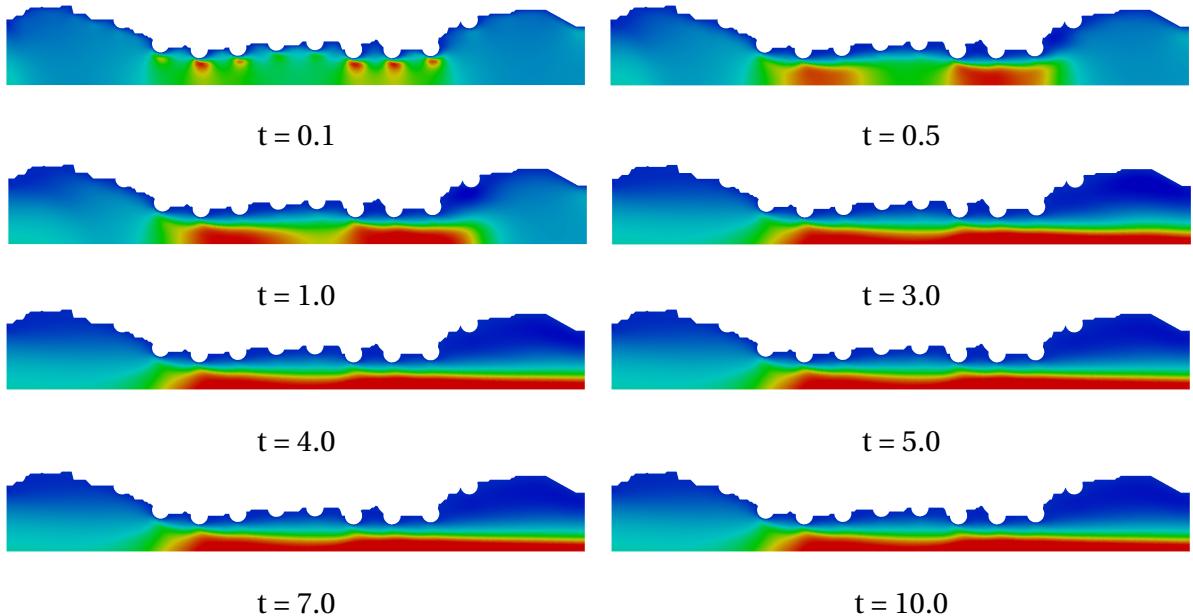


Figure 34: Time and space evolution of the velocity field for real channel with drug-eluting stent.

The Figure 35 and Figure 36 show the time and space evolution of the concentration field for several *Schmidt* number, such as: 1 and 10 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 *Schmidt* number directly influences the drug transport in the blood flow. For high values of the *Schmidt* number, the transport of chemical species becomes purely convective.

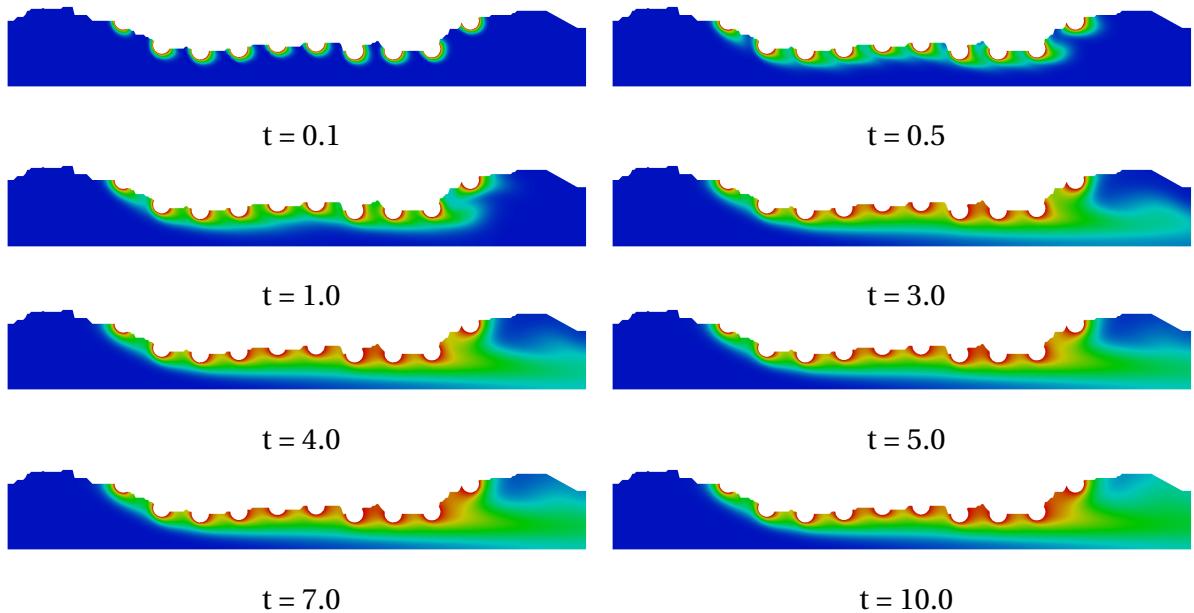


Figure 35: Time and space evolution of the concentration field for real channel with drug-eluting stent when the $Sc = 1$.

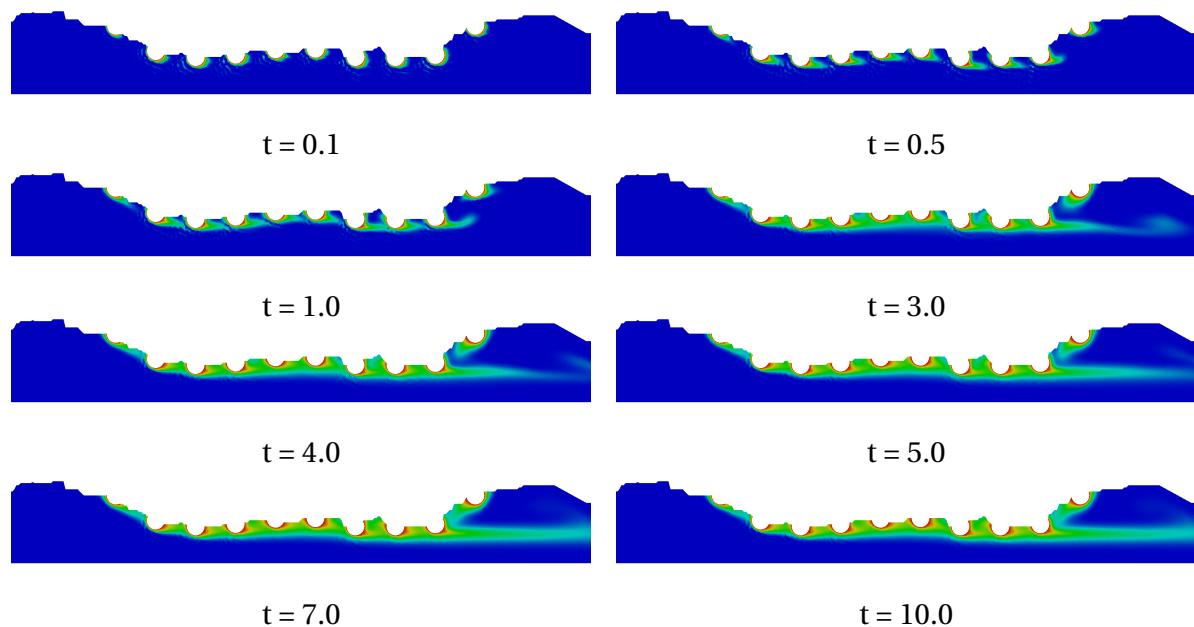


Figure 36: Time and space evolution of the concentration field for real channel with drug-eluting stent when the $Sc = 10$.

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 and the Taylor-Galerkin method was applied to the governing equations. As the vorticity-streamfunction formulation does not present the coupling between velocity and pressure fields, we can use the linear triangular element. In addition to 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 complete code was developed in a high-level programming language using the object orientation paradigm, 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 (as in concentration and temperature) due to the generalized construction of the code.

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. The *lid-driven cavity flow* was also simulated where the results were compared with those presented by Ghia et al. (1982) [8] and Marchi et al. (2009) [9] for several Reynolds number. Finally, the comparison between the *Galerkin* and *Taylor-Galerkin* methods was presented for a purely advection flow of a parabolic function where it was possible to observe the effectiveness of the *Taylor-Galerkin* method compared to *Galerkin* method for decrease spurious oscillations. Thus, the validation of the numerical code was performed for two-dimensional convective-diffusive problems in cartesian coordinates and submitted to the boundary condition of *Dirichlet*.

The objective of this work was to understand the dynamics of blood flow in a coronary artery with atherosclerosis and with a drug-eluting stent. Thus, the simulation for four geometries modeled as two-dimensional and in cartesian coordinates was presented. The profile of the velocity field was shown for the four proposed geometries where it was possible to observe the increase in maximum velocity when the drug-eluting stent was implanted. The simulation was done using several *Schmidt* numbers, such as $Sc = 1$ and 10 . 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 number of *Schmidt*, the transport of chemical

species becomes purely convective and its influence on the artery wall must be verified. In addition to the curved channel shown an acceptable model for the real case, due to the 2% deviation from the maximum non-dimensional velocity.

The following further developments is proposed:

- Utilização do esquema *Semi-Lagrangeano* para as derivadas materiais em substituição do esquema *Taylor-Galerkin* para a redução das oscilações espúrias
- 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

BIBLIOGRAPHY

- [1] JARYL, N. et al. Over-expansion capacity and stent design model: An update with contemporary des platforms. *International Journal of Cardiology*, 2016.
- [2] ORGANIZATION, W. W. H. *Cardiovascular diseases*. 2017. [Online; accessed 03/05/2018 11:55]. Disponível em: <"www.who.int/cardiovascular_diseases/en/">.
- [3] SPRING, B. et al. Better population health through behavior change in adults: a call to action. *Circulation*, 2013.
- [4] GEUZAIN, C.; REMACLE, J. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 2009.
- [5] BABUSKA, I. Error bounds for finite element method. *Numerische Mathematik*, 1971.
- [6] BREZZI, F. On the existence, uniqueness and approximation of saddle-point problems arising from lagrange multipliers. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique.*, 1974.
- [7] PYTHON, S. F. *Python Language Reference, version 2.7*. 1990. [Online; accessed 12/12/2017 12:18]. Disponível em: <<http://www.python.org/>>.
- [8] GHIA, U.; GHIA, K. N.; SHIN, C. T. High-re solutions for incompressible flow using the navier-stokes equations and a multi-grid. *Journal of Computational Physics*, 1982.
- [9] MARCHI, C. H.; SUERO, R.; ARAKI, L. K. The lid-driven square cavity flow: Numerical solution with a 1024 x 1024 grid. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 2009.
- [10] WANG, H. et al. Dynamics of blood flow in coronary artery. *International Congress of Mechanical Engineering*, 2017.
- [11] HENDERSON, A. Paraview guide, a parallel visualization application. *Kitware Inc.*, 2007.
- [12] (WHO), W. W. H. O. The top 10 causes of death. www.who.int/news-room/factsheets/detail/the-top-10-causes-of-death, 2018. [Online; accessed 06-04-20 08:27pm].

- [13] DOTTER, C. T.; JUDKINS, M. P. Transluminal treatment of arteriosclerotic obstruction: Description of a new technic and a preliminary report of its application. *Circulation*, 1964.
- [14] GRÜNTZIG, A. R.; SENNING, A.; SIEGENTHALER, W. E. Nonoperative dilatation of coronary-artery stenosis. *The New England Journal of Medicine*, 1979.
- [15] SIGWART, U. et al. Intravascular stents to prevent occlusion and restenosis after transluminal angioplasty. *The New England Journal of Medicine*, 1987.
- [16] SERRUYS, P. et al. A comparison of balloon-expandable-stent implantation with balloon angioplasty in patients with coronary artery disease. *The New England Journal of Medicine*, 1994.
- [17] HWANG, C.; WU, D.; EDELMAN, E. R. Physiological transport forces govern drug distribution for stent-based delivery. *Circulation*, 2001.
- [18] ZUNINO, P. et al. Numerical simulation of drug eluting coronary stents:mechanics, fluid dynamics and drug release. *Computer Methods in Applied Mechanics and Engineering*, 2009.
- [19] BOZSAK, F.; J-M., C.; BARAKAT, A. Modeling the transport of drugs eluted from stents: physical phenomena driving drug distribution in the arterial wall. *Biomech Model Mechanobiol*, Springer, v. 13, p. 327–347, 2014.
- [20] BUKAC, M. et al. Fluid–structure interaction between pulsatile blood flow and a curvedsented coronary artery on a beating heart: A four stent computationalstudy. *Computer Methods in Applied Mechanics and Engineering*, 2016.
- [21] LUCENA, R. et al. On the transport through polymer layer and porous arterial wall in drug-eluting stents. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 2018.
- [22] GUDINO, E.; CASSIO, M.; SEQUEIRA, A. Influence of non-newtonian blood flow models on drug deposition in thearterial wall. *Journal of Non-Newtonian Fluid Mechanics*, 2019.
- [23] MORICE, M. C. et al. Study with the sirolimus coarted bx velocity balloon-expandable stent in the treatment of patients with de novo native coronary artery lesions. a randomized

- comparison of a sirolimus-elution stent with a standard stent for coronary revascularization. *The New England Journal of Medicine*, 2002.
- [24] GRUBE, E. et al. Taxus i: Six and twelve month results from a randomized, double-blind trial on a slow-release paclitaxel-eluting stent for de novo coronary lesions. *Circulation*, 2003.
- [25] COLOMBO, A. et al. Randomized study to assess the effectiveness of slow and moderate-release polymer based paclitaxel-eluting stents for coronary artery lesions. *Circulation*, 2003.
- [26] SCHAMPAERT, E. et al. Canadian study of the sirolimus-eluting stent in the treatment of patients with long de novo lesions in small native coronary arteries (c-sirius). *Journal of the American College of Cardiology*, 2004.
- [27] ARDISSINO, D. et al. Sirolimus-eluting vs uncoated stents for prevention of restenosis in small coronary arteries. *The Journal of the American Medical Association*, 2004.
- [28] RITZ, W. Über eine neue methode zur lösung gewisser variationsprobleme der mathematischen physik. *Journal für die reine und angewandte Mathematik*, De Gruyter, Berlin, Boston, v. 1909, n. 135, p. 1 – 61, 1909. Disponível em: <<https://www.degruyter.com/view/journals/crll/1909/135/article-p1.xml>>.
- [29] GALERKIN, B. G. Series solution of some problems of elastic equilibrium of rods and plates. *Vestnik inzhenerov i tekhnikov*, v. 19, n. 7, p. 897–908, 1915.
- [30] COURANT, R. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*, 1943.
- [31] ZIENKIEWICZ, O. C.; CHEUNG, Y. K. Finite elements in the solution of fiel problems. *The Engineer*, 1965.
- [32] SPALDING, D. B. A novel finite difference formulation for differential expressions involving both first and second derivatives. *International Journal for Numerical Methods in Engineering*, 1972.
- [33] CHRISTIE, I. et al. Finite element methods for second order differential equations with significant first derivatives. *International Journal for Numerical Methods in Engineering*, 1976.

- [34] HEINRICH, J. C.; HUYAKORN, P. S.; ZIENKIEWICZ, O. C. An upwind finite element scheme for two-dimensional convective transport equation. *International Journal for Numerical Methods in Engineering*, 1977.
- [35] BROOKS, A. N.; HUGHES, T. J. Streamline upwind petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 1982.
- [36] PIRONNEAU, O. On the transport-diffusion algorithm and its applications to the navier-stokes equation. *Numerische Mathematik*, 1982.
- [37] DONEA, J. A taylor-galerkin method for convective transporte problems. *International Journal for Numerical Methods in Engineering*, 1984.
- [38] LOHNER, R.; MORGAN, K.; ZIENKIEWICZ, O. C. The solution of non-linear hyperbolic equation systems by the finite element method. *International Journal of Numerical Methods in Fluids*, 1984.
- [39] DONEA, J.; QUARTAPELLE, L. An introduction to finite element methods for transient advection problems. *Computer Methods in Applied Mechanics and Engineering*, 1992.
- [40] ANJOS, G.; MAGIAVACCHI, N.; THOME, J. An ale-fe method for two-phase flows with dynamics boundaries. *Computer Methods in Applied Mechanics and Engineering*, 2020.
- [41] HIRT, C. W.; AMSDEN, A. A.; COOK, J. An arbitrary lagrangian-eulerian computing method for all flow speeds. *Journal of computational physics*, Elsevier, v. 14, n. 3, p. 227–253, 1974.
- [42] DONEA, J.; GIULIANI, S.; HALLEUX, J.-P. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Computer methods in applied mechanics and engineering*, Elsevier, v. 33, n. 1-3, p. 689–723, 1982.
- [43] DONEA, J. et al. Arbitrary lagrangian–eulerian methods. In *Encyclopedia of Computational Mechanics*, 2004.
- [44] HUGHES, T. J.; LIU, W. K.; ZIMMERMANN, T. K. Lagrangian-eulerian finite element formulation for incompressible viscous flows. *Computer methods in applied mechanics and engineering*, Elsevier, v. 29, n. 3, p. 329–349, 1981.

- [45] BATCHELOR, G. An introduction to fluid dynamics. *Cambridge University Press*, 1967.
- [46] FOX, R. W.; MCDONALD, A. T.; PRITCHARD, P. J. *Introduction to Fluid Mechanics*. [S.l.]: John Wiley and Sons, Inc, 2012.
- [47] PONTES, J.; MANGIAVACCHI, N. *Fenômenos de Transferência com Aplicações às Ciências Físicas e à Engenharia. Volume 1: Fundamentos*. [S.l.]: Sociedade Brasileira de Matemática (SBM), 2016.
- [48] ZIENKIEWICZ, O. C.; TAYLOR, R. L. Finite element method - volume 3: Fluid dynamics. *Butterworth-Heinemann*, 2000.
- [49] HUGHES, T. J. R. *The Finite Elements Method: Linear Static and Dynamic Finite Element Analysis*. [S.l.]: Prentice-Hall, Inc., 2000.
- [50] FISH, J.; BELYTSCHKO, T. *A First Course in Finite Elements*. [S.l.]: John Wiley & Sons, Ltd, 2007.
- [51] ANJOS, G. R. *Hydrodynamics Field Solution of electrochemical Cells Through Finite Element Method*. [S.l.]: Metallurgical and Materials Engineering, Federal University of Rio de Janeiro, Brazil, 2007.
- [52] BRENNER, S. C.; SCOTT, L. R. *The Mathematical Theory of Finite Element Methods*. [S.l.]: Springer, 1994.
- [53] FINLAYSON, B. A. *The Method Weighted Residuals and Variational Principles*. [S.l.]: Elsevier, 1972.
- [54] JONES, E.; OLIPHANT, T.; PETERSON, P. *Scipy - Open source scientific tools for Python*. 2001. [Online; accessed 15/03/2018 09:14]. Disponível em: <<http://www.scipy.org/>>.
- [55] LEWIS, R. W.; NITHIARASU, P.; SEETHARAMU, K. N. *Fundamentals of the Finite Element Method for Heat and Fluid Flow*. [S.l.]: Wiley John and Sons, 2004.
- [56] COMMUNITY, T. S. *numpy.copy function*. [Online; accessed 15/03/2018 16:15]. Disponível em: <<https://docs.scipy.org/doc/numpy/reference/generated/numpy.copy.html>>.
- [57] KESSLER, W. et al. Assessment of coronary blood flow in humans using phase difference mr imaging comparison with intracoronary doppler flow measurement. *International Journal of Cardiac Imaging*, 1998.

- [58] LUCENA, R. et al. Transport through polymer layer and porous arterial wall with binding in drug-eluting stents using the fem. *International Congress of Mechanical Engineering*, 2017.

APÊNDICE - PUBLICAÇÕES

Durante a execução deste estudo, alguns trabalhos foram publicados ou aceitos na forma de artigos em congressos. Este apêndice traz três desses trabalhos. Inicialmente é apresentado o trabalho aceito para publicação pelo *17th Brazilian Congress of Thermal Sciences and Engineering (ENCIT)* que acontecerá em Novembro de 2018. Em seguida, apresentamos o trabalho publicado pelo *X Congresso Nacional de Engenharia Mecânica (CONEM)* com identificação doi://10.26678/ABCM.CONEM2018.CON18-1227 que ocorreu em Maio de 2018. Por fim, é apresentado o trabalho publicado pelo *II Congresso Brasileiro de Fluidodinâmica Computacional (CBCFD)* que ocorreu em Junho de 2018. Abaixo segue um pequeno resumo desses trabalhos:

BLOOD FLOW DYNAMICS SIMULATION IN CORONARY ARTERY WITH DRUG-ELUTING STENT USING FINITE ELEMENT METHOD

17th Brazilian Congress of Thermal Sciences and Engineering (ENCIT 2018)

Novembro/2018 - Aceito para a publicação

The present work aims at developing a computational framework to simulate coronary artery flows in cartesian coordinates. The Finite Element Method (FEM) is used to solve the governing equations of the blood flow in coronary artery with drug-eluting stent placed. The blood was modeled as single-phase, incompressible and newtonian fluid. The Navier-Stokes equation is shown according to the stream-vorticity formulation with coupled species transport equation. The Taylor-Galerkin scheme were used to decrease spurious oscillations as seen for moderate to high Reynolds number. The code proved to be effective by results presented in validation cases. The dynamics of blood flow was shown to a coronary artery with atherosclerosis and drug-eluting stent placed. Therefore, the streamfunction and vorticity formulation showed an useful approximation for to calculate the velocity and concentration fields since the variables are scalars allowing then a smooth implementation.

A NUMERICAL SIMULATION OF BLOOD FLOW DYNAMICS IN CORONARY ARTERY USING STREAMFUNCTION-VORTICITY FORMULATION

X Congresso Nacional de Engenharia Mecânica (CONEM 2018)

Maio/2018 - Publicado

doi://10.26678/ABCM.CONEM2018.CON18-1227

The present work aims at developing a computational framework to simulate coronary artery flows in cartesian coordinates. An accurate method capable of capturing the flow dynamics is strictly required. In this paper a Finite Element Method (FEM) is used to solve the governing equations of the motion of the blood flow found in coronary artery as incompressible fluid using the stream-vorticity formulation with coupled mass transport. The validation of the numerical solution was done by well-known benchmark lid-driven cavity problem and the results were compared with others authors as well as the Hagen-Poiseuille flow for the case straight channel that was compared with analytical solution. The streamfunction and vorticity formulation showed an useful approximation for to calculate the velocity and concentration fields since the variables are scalars allowing then a smooth implementation.

BLOOD FLOW SIMULATION USING STREAM FUNCTION-VORTICITY FEM FORMULATION

II Congresso Brasileiro de Fluidodinâmica Computacional (CBCFD 2018)

Junho/2018 - Publicado

The present work aims at developing a computational framework to simulate coronary artery flows in cartesian coordinates. An accurate method capable of capturing the flow dynamics is strictly required. In this paper a Finite Element Method (FEM) is used to solve the governing equations of the motion of the blood flow found in coronary artery as incompressible fluid using the stream-vorticity formulation with coupled species transport equation. The results were shown for two-dimensional domain in complex geometries of modeled coronary artery channel. The numerical simulation was performed using the streamfunction and vorticity formulation with coupled species transport equation by finite element method approach. The streamfunction and vorticity formulation showed a smooth implementation for to calculate the variables since they are scalars. However, there is a significant difference between the results shown in cartesian coordinates and those shown by Wang et al. (2017) in axisymmetric coordinates.

ENCIT-2018-0357

BLOOD FLOW DYNAMICS SIMULATION IN CORONARY ARTERY WITH DRUG-ELUTING STENT USING FINITE ELEMENT METHOD

Leandro Marques

Gustavo Anjos

Jose Pontes

State University of Rio de Janeiro - UERJ, R. Fonseca Teles 524, 20550-013, Rio de Janeiro, Brazil

marquesleandro67@gmail.com, gustavo.anjos@uerj.br, jose.pontes@uerj.br

Abstract: The present work aims at developing a computational framework to simulate coronary artery flows in cartesian coordinates. The Finite Element Method (FEM) is used to solve the governing equations of the blood flow in coronary artery with drug-eluting stent placed. The blood was modeled as single-phase, incompressible and newtonian fluid. The Navier-Stokes equation is shown according to the stream-vorticity formulation with coupled species transport equation. The Taylor-Galerkin scheme were used to decrease spurious oscillations as seen for moderate to high Reynolds number.

Keywords: Stream-Vorticity Function, Finite Element Method, Taylor-Galerkin Method, Drug-Eluting Stent, Hemodynamics.

1. INTRODUCTION

According to the World Health Organization (2017), more people die annually from the cardiovascular diseases (CVDs) than from any other cause in the world. An estimated 17.7 million people died from CVDs in 2015, representing 31% of all global deaths. About 41% of these deaths were due to coronary artery disease (CAD). The leading cause of the CAD is atherosclerosis where the diameter of the vessel is decreased. Two treatments can be performed: coronary artery bypass grafting (CABG) or percutaneous transluminal coronary angioplasty (PTCA). The PTCA is a minimally invasive procedure where a small wire tube, called stents, is placed. This work aims to develop a Finite Element code for stream-vorticity formulation coupled species transport equation and to know how the dynamics of blood flow in coronary artery with atherosclerosis and with stents struts placed.

The dynamics of blood flow in coronary artery and possible influence of stents struts with computational fluid dynamics (CFD) requires a robust numerical method to compute the solution of the differential equations in a relevant model. The equations that govern the dynamics of blood flow in a coronary artery were developed according to continuum media assumption. Thus, the universal conservation laws such as conservation of mass, conservation of momentum and conservation of species transport were used. The blood was modeled as single-phase, incompressible and newtonian fluid, the diffusion coefficient was considered as constant. The Navier-Stokes equation is shown according to the 2D stream-vorticity formulation with coupled species transport equation in a Finite Element Method approach.

The domain was discretized on an unstructured triangular mesh using the *GMSH* open source as proposed by Geuzaine and Remacle (2009). According to decoupling between velocity field and pressure field achieved by stream-vorticity formulation, the linear triangular element was used. The equations were discretized in time by Taylor series expansion remaining the second order terms to decrease spurious oscillations as seen for moderate to high Reynolds number. Then, the Galerkin formulation was used to discretize in space. Therefore, the Taylor-Galerkin scheme was used as proposed by Donea (1984).

The computational development was done in *Python* language using object-oriented programming paradigm with the aim of reusability and further development. The code validation was made by comparison numerical solution and analytical solution of the *Poiseuille* flow. The comparison of velocity field was done for lid-driven cavity flow with those shown by Ghia *et al.* (1982) and Marchi *et al.* (2009). The dynamics of blood flow and species transport in coronary artery was investigated in 2 test cases as suggested by Wang *et al.* (2017), however modified for 2D cartesian coordinates. The simulation was shown using *Paraview* open source as proposed by Henderson (2007).

2. MATHEMATICAL MODEL

A 2-dimensional Finite Element Method approach is employed to analyse the dynamics of blood flow in coronary artery with atherosclerosis and possible influence of stents struts. The governing equations were developed according to continuum media assumption. Thus, the universal conservation laws such as conservation of mass, conservation of momentum and conservation of species transport were used. The blood was modeled as single-phase, incompressible and newtonian fluid, the diffusion coefficient was considered as constant. The Navier-Stokes equation is shown according to stream-vorticity formulation with coupled species transport equation.

$$\frac{\partial \omega}{\partial t} + \mathbf{v} \cdot \nabla \omega = \frac{1}{Re} \nabla^2 \omega \quad (1)$$

$$\nabla^2 \psi = -\omega \quad (2)$$

$$\mathbf{v} = \mathbf{D}\psi \quad (3)$$

$$\frac{\partial c}{\partial t} + \mathbf{v} \cdot \nabla c = \frac{1}{ReSc} \nabla^2 c \quad (4)$$

Where, ω is the vorticity field, ψ is the stream function field, c is the concentration field, $\mathbf{v} = (u, v)$ is the velocity field, $\mathbf{D} = [\partial/\partial y, -\partial/\partial x]$ is a mathematical operator, $Re = \rho u D / \mu$ is the Reynolds number, $Sc = \nu / D$ is the Schmidt number, x and y are the independent spatial variables and t is the time variable.

2.1 Finite Element Method

The domain was discretized on an unstructured triangular mesh using the *GMSH* open source as proposed by Geuzaine and Remacle (2009). According to decoupling between velocity field and pressure field achieved by stream-vorticity formulation, the use of linear triangular element was used. The equations were discretized in time by Taylor series expansion remaining the second order terms to decrease spurious oscillations as seen for moderate to high Reynolds number. Then, the Galerkin formulation was used to discretize in space. Therefore, the Taylor-Galerkin scheme was used as proposed by Donea (1984). The governing equations in matrix form used in this paper were:

$$\left[\frac{M}{\Delta t} + \frac{1}{Re} [K_{xx} + K_{yy}] \right] \omega^{n+1} = \frac{M}{\Delta t} \omega^n - u \cdot G_x \omega^n - v \cdot G_y \omega^n - u \frac{\Delta t}{2} [u K_{xx} + v K_{yx}] \omega^n - v \frac{\Delta t}{2} [u K_{xy} + v K_{yy}] \omega^n \quad (5)$$

$$[K_{xx} + K_{yy}] \psi = M \omega \quad (6)$$

$$Mu = G_y \psi \quad (7)$$

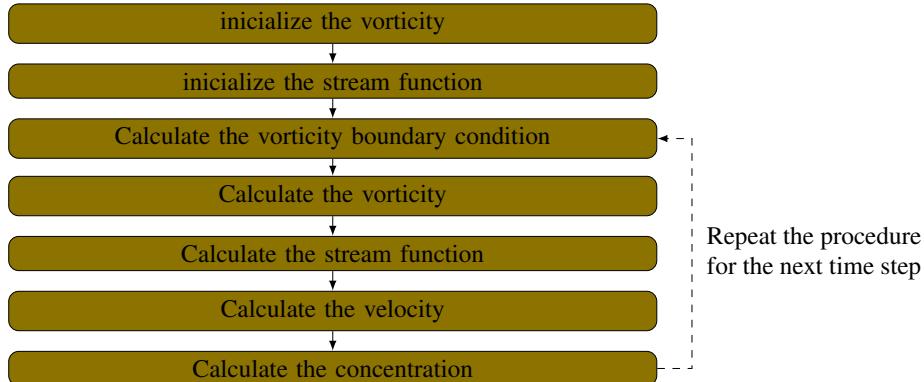
$$Mv = -G_x \psi \quad (8)$$

$$\left[\frac{M}{\Delta t} + \frac{1}{ReSc} [K_{xx} + K_{yy}] \right] c^{n+1} = \frac{M}{\Delta t} c^n - u \cdot G_x c^n - v \cdot G_y c^n - u \frac{\Delta t}{2} [u K_{xx} + v K_{yx}] c^n - v \frac{\Delta t}{2} [u K_{xy} + v K_{yy}] c^n \quad (9)$$

Where, M is mass matrix, G_x and G_y are gradient matrix, K_{xx} , K_{xy} , K_{yx} and K_{yy} are stiffness matrix. The last term of the Eqs. 5 and 9 is known as numerical diffusion and it decrease the spurious oscillations as seen for moderate to high Reynolds numbers. For scalars, *Taylor Galerkin Method* and *Characteristic Galerkin* produce the same result as showed by Lohner *et al.* (1984). The superscripts $n+1$ and n are the scalar that will be calculated and that was calculated in the previous time step, respectively.

2.2 Numerical Solution

The computational development was done in *Python* language using object-oriented programming paradigm with the aim of reusability and further development. The linear system of equations that come from implementing the FEM is solved through iterative method *Conjugate Gradient Solver* available in the public library for scientific tools *SciPy* maintained by Jones *et al.* (2001). The solution algorithm used is shown below:



The first and second steps are out of time loop, while the third to the seventh step are inside of time loop. The application of the boundary condition in the equation can be before loop, except for the vorticity equation (*fourth step*) that the boundary condition must be applied at each time step.

3. VALIDATION

The code validation was made by comparison numerical solution and analytical solution of the *Poiseuille flow*. The comparison of velocity field was done for lid-driven cavity flow with those shown by Ghia *et al.* (1982) and Marchi *et al.* (2009).

3.1 Poiseuille Flow

A single-phase flow, steady and fully developed of an incompressible and newtonian fluid between parallel horizontal plates and stationary is maintained due to a pressure gradient. This flow is known as *Poiseuille flow*. In Fig. ?? is shown a schematic representation of the numerical domain used to simulate the Poiseuille Flow problem, where no-slip condition were used at the top and bottom walls, while an inflow and outflow conditions were set in the plane $x = 0$ and $x = 5L$ respectively.

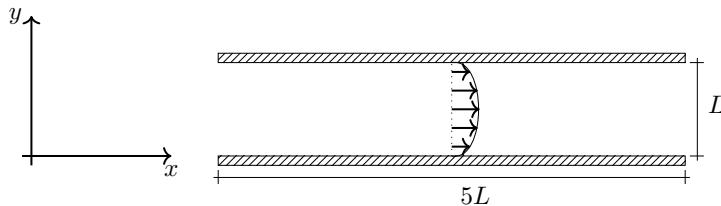


Figure 1: Poiseuille flow

The boundaries conditions used were:

- *inflow condition*: the normal velocity component is set to null value $v = 0$. The tangent velocity component is set to parabolic profile by exact solution, that is, $u = [4u_{max}y/L^2][L - y]$, where $u_{max} = 1.5$. The streamfunction is also specified and its value is defined according to continuity equation for an incompressible fluid. Thus, its value is $\psi = [2u_{max}y^2/3L^2][3L - 2y]$.
- *No-slip condition*: all the velocity components are specified with null value $u = 0$ and $v = 0$. The streamfunction is also specified $\psi = 1$ in top plate and $\psi = 0$ in bottom plate.

- *outflow condition*: no value is specified. The derivatives of the tangent velocity component, of the normal velocity component and of streamfunction are set to null values, that is, $\partial u/\partial n = 0$, $\partial v/\partial n = 0$ and $\partial \psi/\partial n = 0$ respectively.

The velocity field profile is calculated by equation below:

$$u = \frac{4u_{max}}{L^2}y[L - y] \quad (10)$$

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

The Fig. 2 shown the evolution of velocity field profile in time when $Re = 100$, the comparasion is also shown between numerical solution and analytical solution. Therefore, the code shown a satisfactory result.

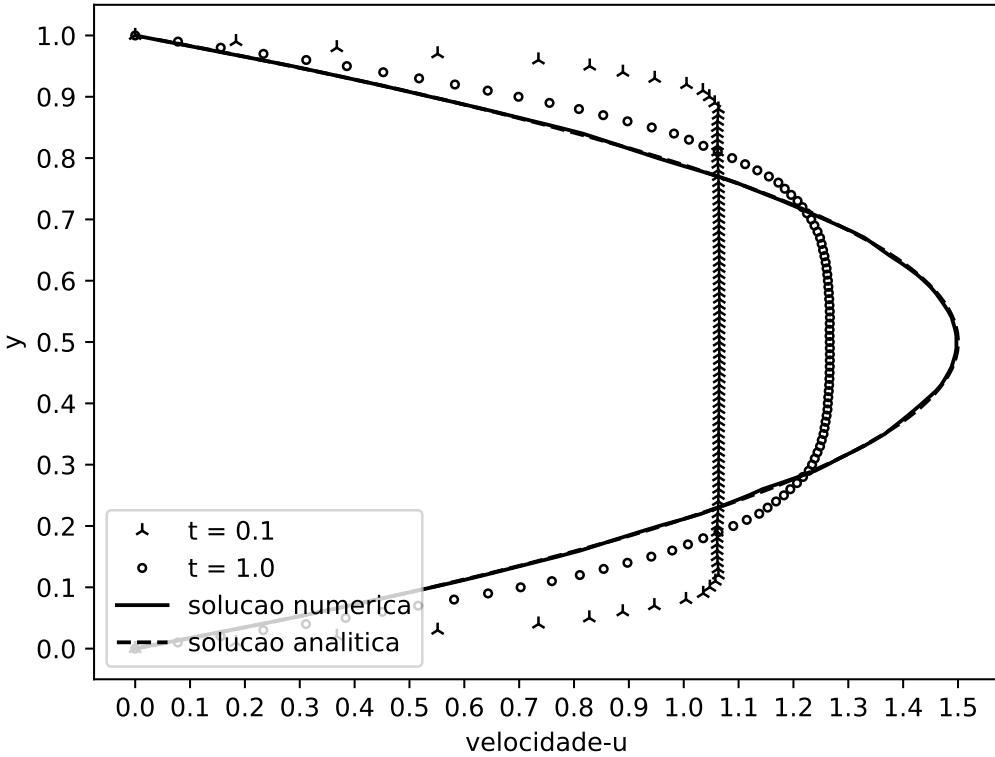


Figure 2: Evolution of velocity field profile in time for $Re = 100$ and the comparison between numerical solution and analytical solution.

3.2 Lid-driven Cavity Flow

A flow on cavity when side and bottom plates are stationary and top plate moves with velocity constant such that $U = top = 1$ is known as *lid-driven cavity flow*. In Fig. ?? is shown a schematic representation of the numerical domain used to simulate the Lid-driven Cavity Flow problem, where no-slip condition were used at the bottom and side walls, while at top wall, the velocity was set $u = U_{top}$.

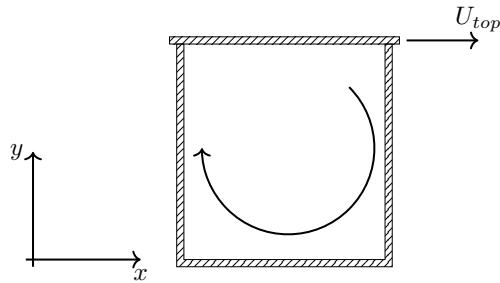


Figure 3: Lid-driven cavity flow

Were simulated flows for the following Reynolds numbers: (Re): 10, 100, 400 and 1000. The dimensions domain in x-direction and y-direction are [0,1]. We used a mesh with 1563 nodes and 2988 elements

The boundaries conditions used were:

- *top plate moves*: all the velocity components are specified with $v = 0$ and $u = U_{top}$, where $U_{top} = 1$. The streamfunction is also specified and its value is $\psi = 0$.
- *No-slip condition*: This condition is used on side and bottom plates. All the velocity components are also specified with null value $u = 0$ and $v = 0$. The streamfunction is specified as $\psi = 0$.

The Figs. 4 and 5 shown profile of u and v components respectively for several Reynolds numbers. The results were compared with Ghia *et al.* (1982) and Marchi *et al.* (2009). Therefore, the code shown a satisfactory result.

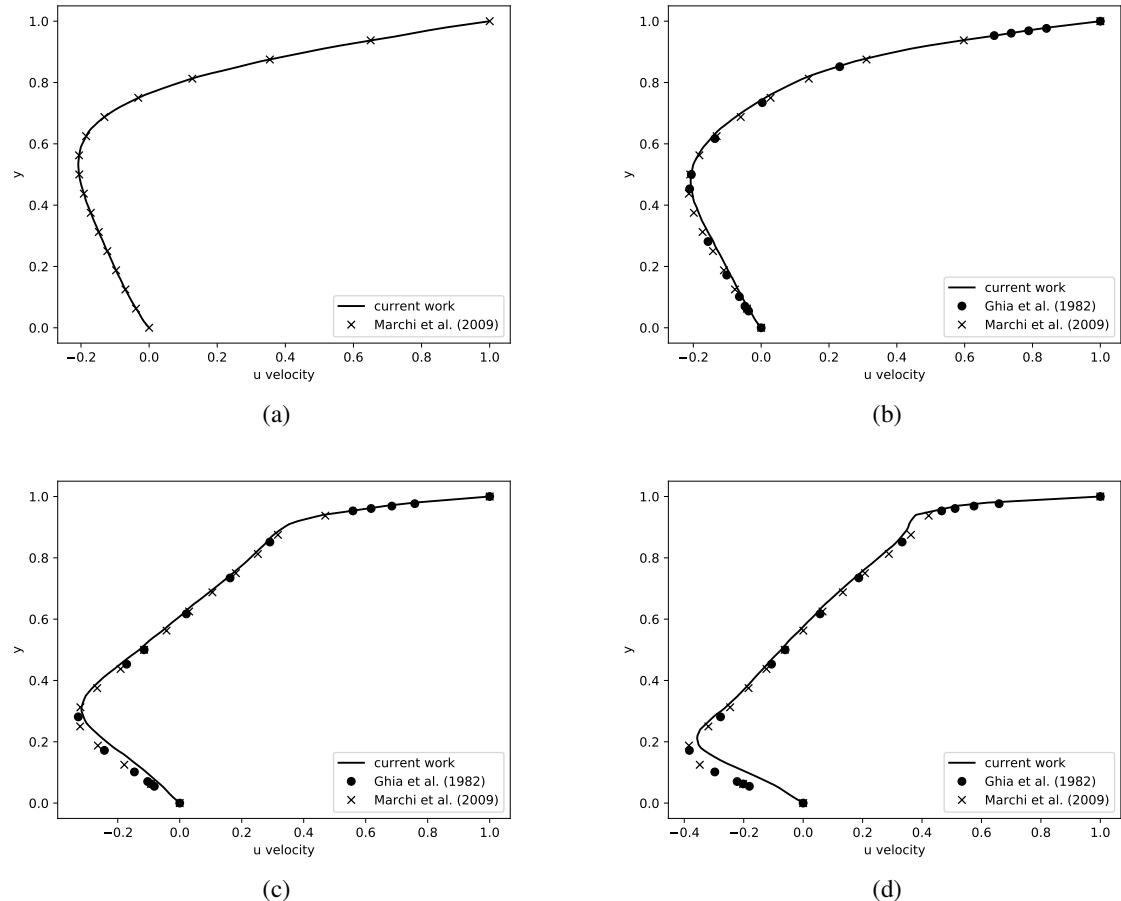


Figure 4: Centerline u velocity profile ($x = 0.5$) in a lid-driven cavity for different Reynolds numbers: (a) 10 (b) 100 (d) 400 (f) 1000.

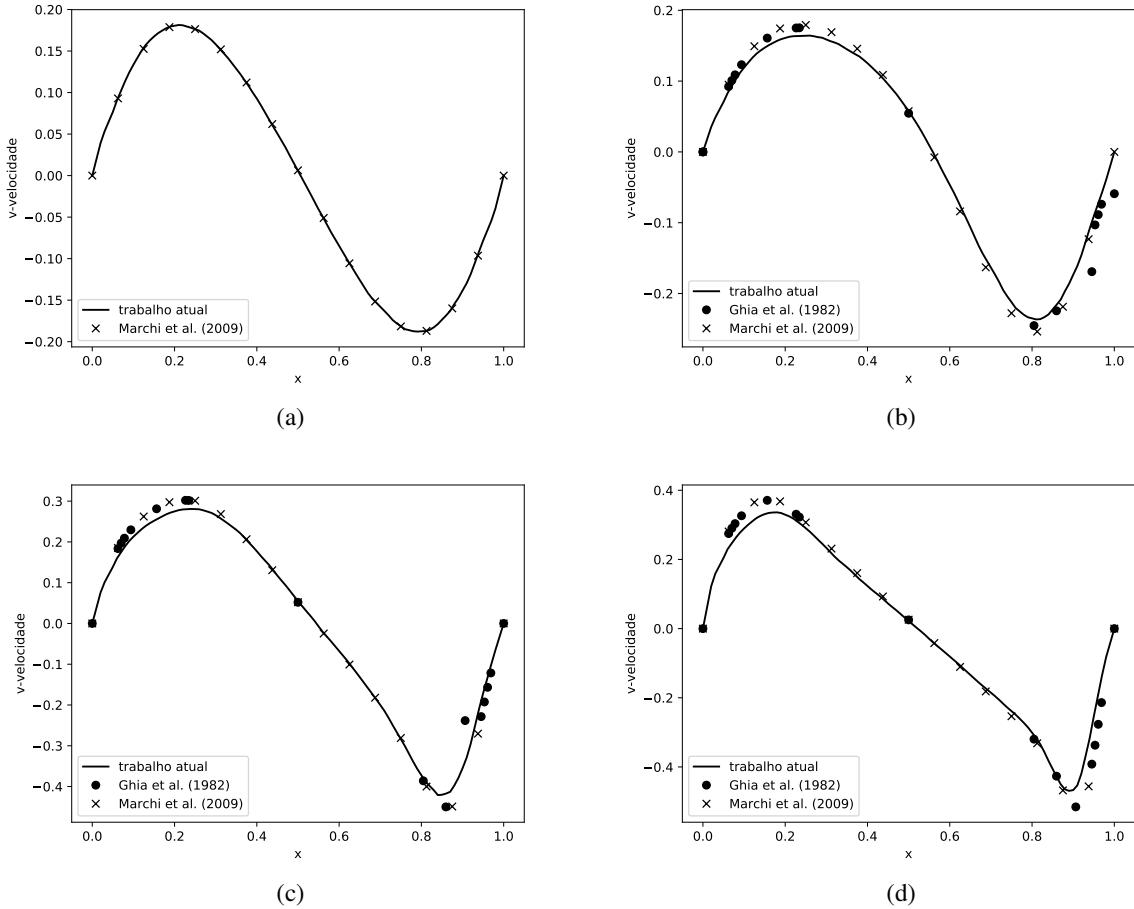


Figure 5: Centerline v velocity profile ($y = 0.5$) in a lid-driven cavity for different Reynolds numbers: (a) 10 (b) 100 (d) 400 (f) 1000.

4. RESULTS AND DISCUSSION

Some results of simulations are shown to demonstrate its capability of using unstructured triangular meshes on various geometries and combination of geometries. Numerical results are given for several cases of blood flows in artery when $Sc = 10$. The post-processing was performed by open source software *PARAVIEW* proposed by Henderson (2007). The lumen radius of a typical artery is about $R = 0.0015\text{m}$, viscosity in the lumen are set to $\mu = 0.0035\text{Pa.s.}$ and density $\rho = 1060\text{kg/m}^3$ as suggested by Bozsak *et al.* (2014). According to Kessler *et al.* (1998), the velocity of the flow at coronary artery is $v = 12\text{cm/s.}$ Therefore, the Reynolds number is $Re = 54.5.$

The Navier-Stokes equation is used according to stream-vorticity formulation with species transport equation coupled for 2 different geometries proposed by Wang *et al.* (2017) and is shown in the Fig. 6, however modified for cartesian coordinates. According to symmetry on y coordinate, half domain was simulated.

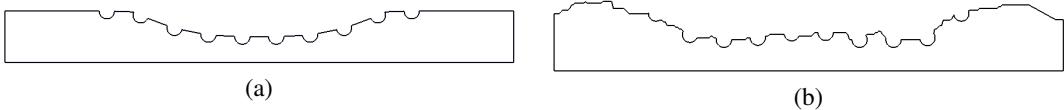


Figure 6: Non-dimensional geometry for blood flow dynamics in coronary arteries. The channel length $L = 10R$ is based on the channel width $R = 1.$ (a) Curved Channel with Stent and (b) Real Channel with Stent.

The boundaries conditions used were:

- *inflow condition:* the normal velocity component is set to null value $v = 0.$ The tangent velocity component is set

to parabolic profile by exact solution, that is, $u = u_{max} [1 - (y/L)^2]$, where $u_{max} = 1.5$. The streamfunction is also specified and its value is defined according to continuity equation for an incompressible fluid. Thus, its value is $\psi = [u_{max}y/3] [3 - (y/L)^2]$.

- *No-slip condition*: this condition is used on bottom plate. all the velocity components are specified with null value $u = 0$ and $v = 0$. The streamfunction is also specified $\psi = 1$ and the derivative of concentration has null value $\partial c/\partial n = 0$.
- *outflow condition*: no value is specified. The derivatives of the tangent velocity component, of the normal velocity component and of streamfunction are set to null values, that is, $\partial u/\partial n = 0$, $\partial v/\partial n = 0$, $\partial \psi/\partial n = 0$ and $\partial c/\partial n = 0$ respectively.
- *Free-slip condition*: used when a symmetry condition is desired. The normal velocity component is set to null value $v = 0$ as well as the streamfunction $\psi = 0$. The derivative of the tangent velocity component and the derivative of the concentration are also set to null value $\partial u/\partial n = 0$ and $\partial c/\partial n = 0$ respectively.
- *Strut condition*: used on the stent. The normal velocity component and the tangent velocity component are specified with null value $u = 0$ and $v = 0$. The streamfunction and the concentration are also specified $\psi = 1$ and $c = 1$ respectively.

4.1 Curved Channel with Stent

For the case when coronary artery has atherosclerosis and drug-eluting stent is placed, the problem is modeled as a parallel and curved plates flow and the stent is modeled by 10 semi-circles uniformly spaced. The geometry used promotes a smooth reduction of length between the bottom and top plates. Were considered 40% of channel obstruction due to atherosclerosis. The Fig. 10 shown velocity field profile along y coordinates in centerline ($x = 5R$).

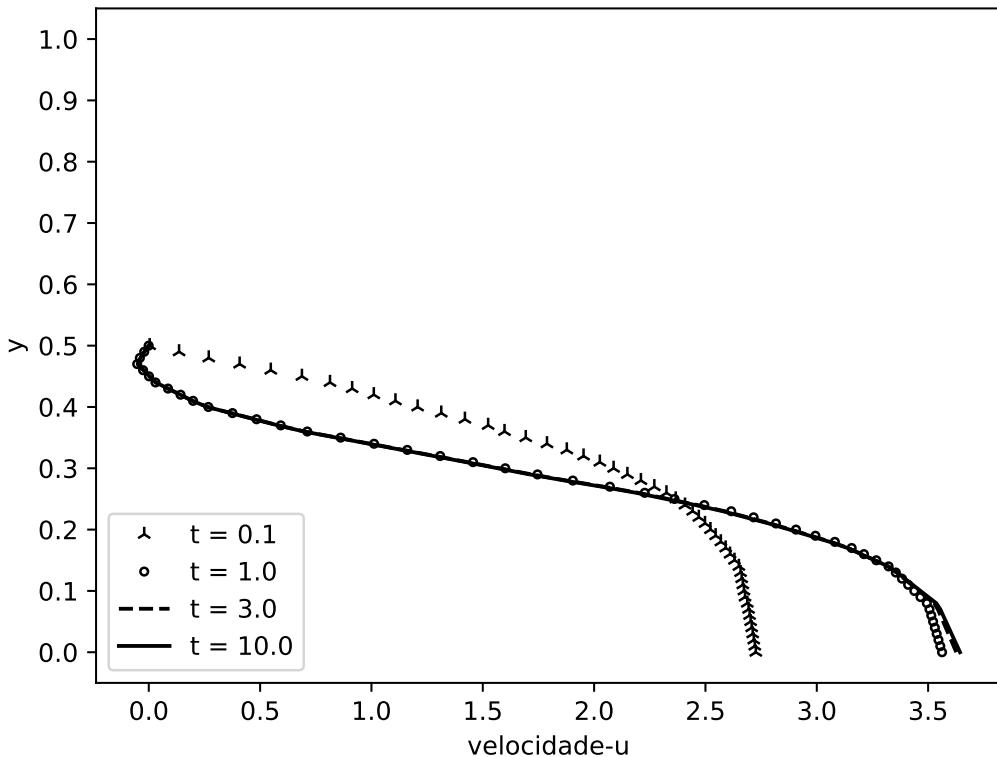


Figure 7: Evolution of velocity profile in time for Curved Channel with Drug-eluting Stent.

In Figs 8 and 9, are shown the velocity and concentration evolution in time and space for half domain according to symmetry y coordinate. The concentration field is represented with non-dimensional values when the red color is $c = 1$ and blue color $c = 0$.

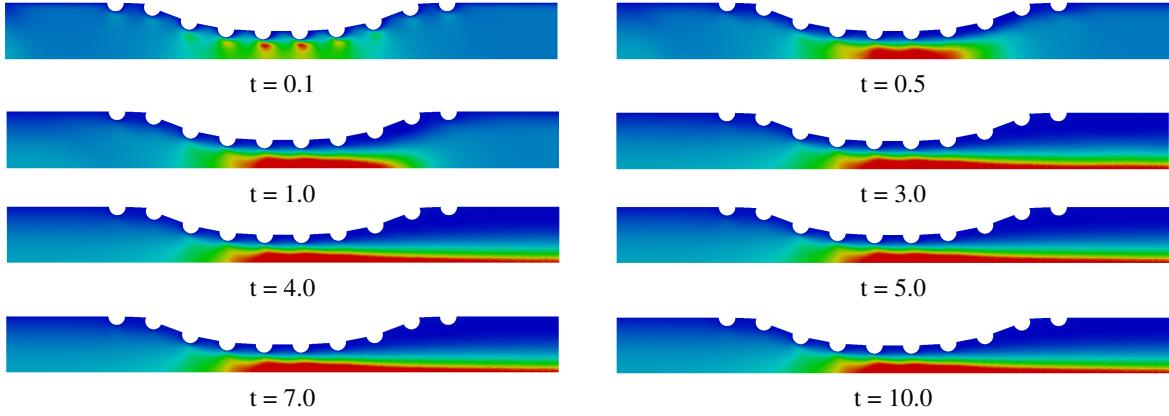


Figure 8: Evolution in time and space of velocity field for Curved Channel with Drug-eluting Stent.

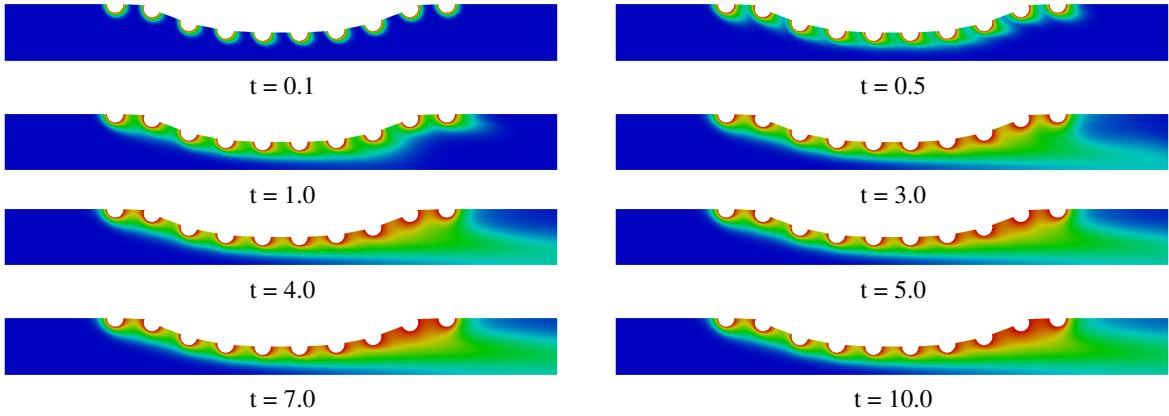


Figure 9: Evolution in time and space of concentration field for Curved Channel with Drug-eluting Stent.

4.2 Real Channel with Stent

In this case, the real coronary artery with atherosclerosis and drug-eluting stent placed is performed. The geometry was taken using image processing from a real coronary artery photography. The stent is modeled by 10 semi-circles uniformly spaced. As in other case, the geometry used promotes a reduction of length between the bottom and top plates when were considered 40% of channel obstruction due to atherosclerosis. The Fig. 10 shown velocity field profile along y coordinates in centerline ($x = 5R$).

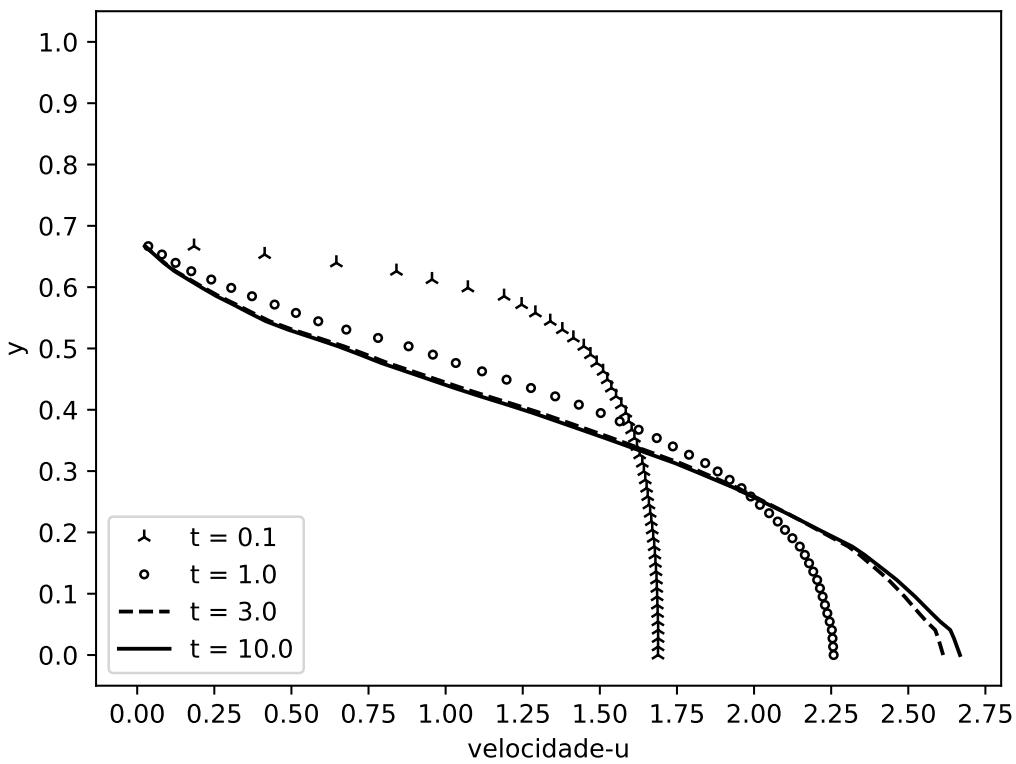


Figure 10: Evolution of velocity profile in time for Curved Channel with Drug-eluting Stent.

In Figs 11 and 12, are shown the velocity and concentration evolution in time and space for half domain according to symmetry y coordinate. The concentration field is represented with non-dimensional values when the red color is $c = 1$ and blue color $c = 0$.

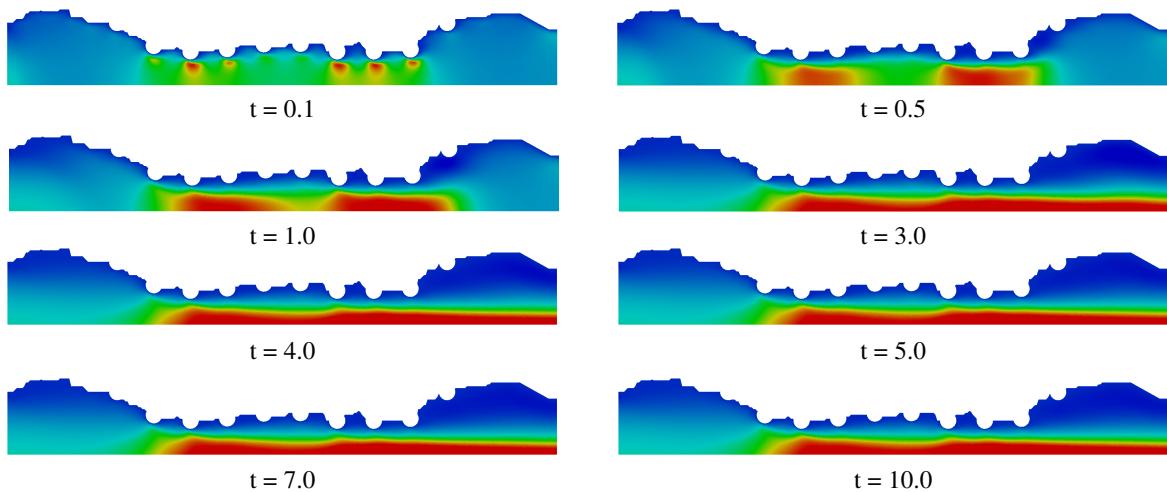


Figure 11: Evolution in time and space of velocity field for Real Channel with Drug-eluting Stent.

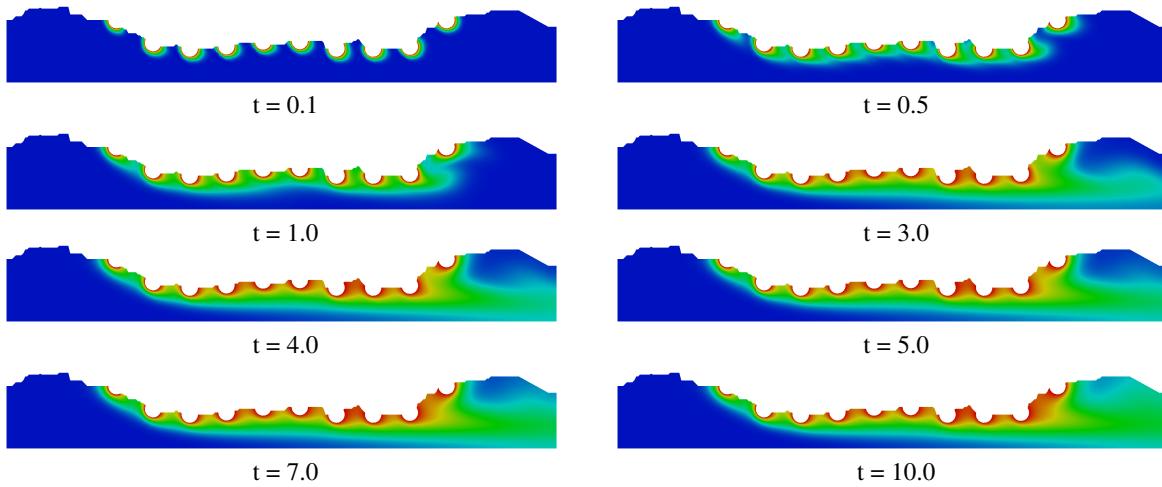


Figure 12: Evolution in time and space of concentration field for Real Channel with Drug-eluting Stent.

5. CONCLUSION

In this work, a numerical code for Navier-Stokes equation according to stream-vorticity formulation with species transport equation coupled was developed using Finite Element Method. The Taylor-Galerkin scheme was applied to decrease the spurious oscillations as seen for moderate to high Reynolds number. The code proved to be effective by results presented in validation cases. The dynamics of blood flow was shown to a coronary artery with atherosclerosis and drug-eluting stent placed. Therefore, the streamfunction and vorticity formulation showed an useful approximation for to calculate the velocity and concentration fields since the variables are scalars allowing then a smooth implementation.

6. ACKNOWLEDGEMENTS

The authors thank the FAPERJ (Research Support Foundation of the State of Rio de Janeiro) for its financial support.

7. REFERENCES

- Bozsak, F., J-M., C. and Barakat, A., 2014. "Modeling the transport of drugs eluted from stents: physical phenomena driving drug distribution in the arterial wall". *Biomech Model Mechanobiol*, Vol. 13, pp. 327–347. doi:10.1007/s10237-013-0546-4.
- Donea, J., 1984. "A taylor-galerkin method for convective transport problems". *International Journal for Numerical Methods in Engineering*.
- Geuzaine, C. and Remacle, J., 2009. "Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities". *International Journal for Numerical Methods in Engineering*.
- Ghia, U., Ghia, K.N. and Shin, C.T., 1982. "High-re solutions for incompressible flow using the navier-stokes equations and a multi-grid." *Journal of Computational Physics*.
- Henderson, A., 2007. "Paraview guide, a parallel visualization application". *Kitware Inc.*
- Jones, E., Oliphant, T. and Peterson, P., 2001. "Scipy - open source scientific tools for python". URL <http://www.scipy.org/>. [Online; accessed 15/03/2018 09:14].
- Kessler, W., Moshage, W., Galland, A., Zink, D., Achenbach, S., Nitz, W., Laub, G. and Bachmann, K., 1998. "Assessment of coronary blood flow in humans using phase difference mr imaging comparison with intracoronary doppler flow measurement." *International Journal of Cardiac Imaging*.
- Lohner, R., Morgan, K. and Zienkiewicz, O.C., 1984. "The solution of non-linear hyperbolic equation systems by the finite element method". *International Journal of Numerical Methods in Fluids*.
- Marchi, C.H., Suero, R. and Araki, L.K., 2009. "The lid-driven square cavity flow: Numerical solution with a 1024 x 1024 grid." *Journal of the Brazilian Society of Mechanical Sciences and Engineering*.
- Wang, H., McGinty, S., Lucena, R., Pontes, J., Anjos, G. and Mangiavacchi, N., 2017. "Dynamics of blood flow in coronary artery". *International Congress of Mechanical Engineering*.
- World Health Organization, W., 2017. "Cardiovascular diseases". URL http://www.who.int/cardiovascular_diseases/en/. [Online; accessed 03/05/2018 11:55].

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.



A NUMERICAL SIMULATION OF BLOOD FLOW DYNAMICS IN CORONARY ARTERY USING STREAMFUNCTION-VORTICITY FORMULATION

Leandro Marques, marquesleandro67@gmail.com¹

Gustavo Anjos, gustavo.anjos@uerj.br¹

José Pontes, jose.pontes@uerj.br¹

¹State University of Rio de Janeiro - UERJ, R. Fonseca Teles 524, 20550-013, Rio de Janeiro, Brazil

Abstract: The present work aims at developing a computational framework to simulate coronary artery flows in cartesian coordinates. An accurate method capable of capturing the flow dynamics is strictly required. In this paper a Finite Element Method (FEM) is used to solve the governing equations of the motion of the blood flow found in coronary artery as incompressible fluid using the stream-vorticity formulation with coupled mass transport.

Keywords: Finite Element Method, Taylor-Galerkin Method, Biomedical Engineering, Hemodynamics, Atherosclerosis.

doi://10.26678/ABCM.CONEM2018.CON18-1227

1. INTRODUCTION

According to the World Health Organization (2017), more people die annually from the cardiovascular diseases (CVDs) than from any other cause in the world. An estimated 17.7 million people died from CVDs in 2015, representing 31% of all global deaths. About 41% of these deaths were due to coronary artery disease (CAD). The leading cause of the CAD is atherosclerosis where the diameter of the vessel is decreased. Two treatments can be performed: coronary artery bypass grafting (CABG) or percutaneous transluminal coronary angioplasty (PTCA). The PTCA is a minimally invasive procedure where a small wire tube, called stents, is placed. This work aims to know how the dynamics of blood flow in coronary artery with atherosclerosis and with stents struts placed.

The dynamics of blood flow in coronary artery and possible influence of stents struts with computational fluid dynamics (CFD) requires a robust numerical method to compute the solution of the differential equations in a relevant model. We consider the model of dissolution and transport of sirolimus on a two-dimensional domain representing the polymer coating layer and the hydrodynamic of the blood flow in the artery in the vicinity of a stent strut as suggested by Bozsak *et al.* (2014) and McGinty and Pontrelli (2016). Also, the effect of the releasing process of the polymers is considered. However, the spatial distribution of the sirolimus is greatly influenced by the flow and the arterial wall properties, being therefore susceptible to patient health conditions. The difference of artery shapes and existence of the struts have been investigated in 4 test cases in the influence on the flow dynamics and the transport of a chemical species as suggested by Wang *et al.* (2017).

We employ the Finite Element method on an unstructured mesh created by open source software called *GMSH* proposed by Geuzaine and Remacle (2009) for the discretization of the incompressible single-phase Navier-Stokes through the stream-vorticity function coupled with the concentration equation. We apply the Taylor-Galerkin method to the decrease the spurious oscillations as seen for moderate to high Reynolds numbers as proposed by Donea (1984) and Zienkiewicz and Taylor (2000) and was used the well-known benchmark problem lid-driven cavity for the validation of the numerical code where the results were compared with Ghia *et al.* (1982) and Marchi *et al.* (2009).

2. MATHEMATICAL MODEL

A 2-dimensional Finite Element Method approach is employed to analyse the dynamics of blood flow in coronary artery and possible influence of stents struts. The first step in its development is the initial modelling of the problem, and in this case it can be described by a formulation using the vorticity and stream function applied in the conservation of momentum equation for incompressible flow and the concentration distribution equation. This approach makes the model useful for the cases of single-phase flows.

$$\frac{\partial w}{\partial t} + \mathbf{v} \cdot \nabla w = \frac{1}{Re} \nabla^2 w \quad (1)$$

$$\nabla^2 \psi = -w \quad (2)$$

$$\mathbf{v} = \mathbf{D}\psi \quad (3)$$

$$\frac{\partial c}{\partial t} + \mathbf{v} \cdot \nabla c = \frac{1}{ReSc} \nabla^2 c \quad (4)$$

Where, ω is the vorticity field, ψ is the stream function field, c is the concentration field, $\mathbf{v} = (u, v)$ is the velocity field, $\mathbf{D} = [\partial/\partial y, -\partial/\partial x]$ is a mathematical operator, $Re = \rho u D / \mu$ is the Reynolds number, $Sc = \nu/D$ is the Schmidt number, x and y are the independent spatial variables and t is the time variable.

2.1 Initial and Boundary Conditions

As commented by Anjos (2012), the initial conditions and the boundary conditions are of utmost importance to realistically characterizing any problem modeled by differential equations. Thus, the boundary conditions used in this paper are briefly explained below:

- *inflow condition*: the normal velocity component is set to null value $v = 0$. The derivates of the streamfunction and the concentration are set null values $\partial\psi/\partial n = 0$ and $\partial c/\partial n = 0$ respectively. For the tangent velocity component is set a half parabolic profile comes from the analytical solution of a Straight Channel in the cartesian coordinates: $u = u_{max}(1 - (y/R)^2)$, where $u_{max} = 2$.
- *No-slip condition*: all the velocity components are specified with null value $u = 0$ and $v = 0$. The streamfunction is also specified $\psi = 1.3$. The derivative concentration is set null value $\partial c/\partial n = 0$.
- *outflow condition*: no value is specified. The derivatives of the tangent velocity component, of the normal velocity component, of streamfunction and of concentration are set to null values, that is, $\partial u/\partial n = 0$, $\partial v/\partial n = 0$, $\partial\psi/\partial n = 0$ and $\partial c/\partial n = 0$ respectively.
- *Free-slip condition*: used when a symmetry condition is desired. The normal velocity component is set to null value $v = 0$ as well as the streamfunction $\psi = 0$. The derivative of the tangent velocity component and the derivative of the concentration are also set to null value $\partial u/\partial n = 0$ and $\partial c/\partial n = 0$ respectively.
- *Strut condition*: used on the stent. The normal velocity component and the tangent velocity component are specified with null value $u = 0$ and $v = 0$. The streamfunction and the concentration are also specified $\psi = 1.3$ and $c = 1$ respectively.

The boundary condition of the vorticity field is calculated in the solution algorithm scheme as we will see later.

2.2 Finite Element Method

This is paper, was used the Galerkin formulation to discretize the government equations. The spatial domain was discretized using linear triangular elements by open source software called GMSH proposed by Geuzaine and Remacle (2009). For the discretization of the temporal domain was used a finite differences method with a forward difference approximation and the Taylor-Galerkin Method with omitted terms of higher order than two was used to the decrease the spurious oscillations for moderate to high Reynolds numbers as proposed by Donea (1984) and Zienkiewicz and Taylor (2000). Therefore, the governing equations in matrix form used in this paper were:

$$\left[\frac{M}{\Delta t} + \frac{1}{Re} [K_{xx} + K_{yy}] \right] w^{n+1} = \frac{M}{\Delta t} w^n - u \cdot G_x w^n - v \cdot G_y w^n - u \frac{\Delta t}{2} [u K_{xx} + v K_{xy}] w^n - v \frac{\Delta t}{2} [u K_{yx} + v K_{yy}] w^n \quad (5)$$

$$[K_{xx} + K_{yy}] \psi = M w \quad (6)$$

$$M u = G_y \psi \quad (7)$$

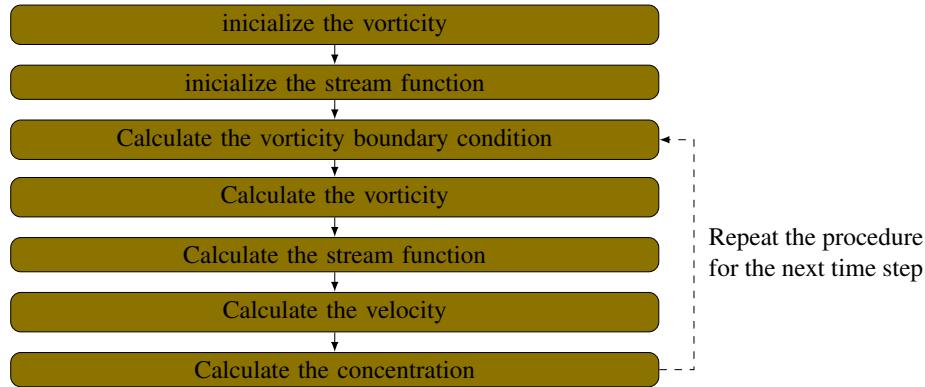
$$Mv = -G_x \psi \quad (8)$$

$$\left[\frac{M}{\Delta t} + \frac{1}{ReSc} [K_{xx} + K_{yy}] \right] c^{n+1} = \frac{M}{\Delta t} c^n - u \cdot G_x c^n - v \cdot G_y c^n - u \frac{\Delta t}{2} [u K_{xx} + v K_{xy}] c^n - v \frac{\Delta t}{2} [u K_{yx} + v K_{yy}] c^n \quad (9)$$

Where, M is mass matrix, G_x and G_y are gradient matrix, K_{xx} , K_{xy} , K_{yx} and K_{yy} are stiffness matrix. The last term of the Eqs. 5 and 9 is known as numerical diffusion and it decrease the spurious oscillations as seen for moderate to high Reynolds numbers. For scalars, *Taylor Galerkin Method* and *Characteristic Galerkin* produce the same result as showed by Lohner *et al.* (1984). The superscripts $n+1$ and n are the scalar that will be calculated and that was calculated in the previous time step, respectively.

2.3 Numerical Solution

The linear system of equations that come from implementing the FEM is solved throught iterative method *Conjugate Gradient Solver* available in the public library for scientific tools *SciPy* maintained by Jones *et al.* (2001) in the Python language. The solution algorithm used is shown below:



that is:

1. Inicialize the vorticity with the equation:

$$Mw = G_x v - G_y u$$

2. Inicialize the stream function with the equation:

$$[K_{xx} + K_{yy}] \psi = Mw$$

It is necessary to apply the boundary condition in the equation.

3. Calculate the vorticity boundary condition with the equation:

$$Mw = G_x v - G_y u$$

4. Calculate the vorticity with the equation:

$$\left[\frac{M}{\Delta t} + \frac{1}{Re} [K_{xx} + K_{yy}] \right] w^{n+1} = \frac{M}{\Delta t} w^n - u \cdot G_x w^n - v \cdot G_y w^n - u \frac{\Delta t}{2} [u K_{xx} + v K_{xy}] w^n - v \frac{\Delta t}{2} [u K_{yx} + v K_{yy}] w^n$$

Where w^n is the vorticity calculated in the previous time step and w^{n+1} is the vorticity that will be calculated in the time step. It is necessary to apply the boundary condition calculated in the third step before finding vorticity field.

5. Calculate the stream function with equation:

$$[K_{xx} + K_{yy}] \psi = Mw$$

It is necessary to apply the boundary condition in the equation before finding streamfunction field.

6. Calculate the velocity with the equation:

$$Mu = G_y \psi$$

$$Mv = -G_x \psi$$

It is necessary to apply the boundary condition in the equation before u and v . Note that one can solve the linear system using the conjugate gradient method, since M is positive and definite matrix or M can be approximated by its lumped version and therefore easily inverted.

7. Calculate the concentration with equation:

$$\left[\frac{M}{\Delta t} + \frac{1}{ReSc} [K_{xx} + K_{yy}] \right] c^{n+1} = \frac{M}{\Delta t} c^n - u \cdot G_x c^n - v \cdot G_y c^n \\ - u \frac{\Delta t}{2} [u K_{xx} + v K_{xy}] c^n - v \frac{\Delta t}{2} [u K_{yx} + v K_{yy}] c^n$$

Where c^n is the vorticity calculated in the previous time step and c^{n+1} is the vorticity that will be calculated in the time step. It is necessary to apply the boundary condition in the equation before finding concentration field.

8. Return the third step and repeat the procedure for next time step.

The first and second steps are out of time loop, while the third to the seventh step are inside of time loop. The application of the boundary condition in the equation can be before loop, except for the vorticity equation (*fourth step*) that the boundary condition must be applied at each time step.

3. VALIDATION

The validation of the numerical simulation was done by well-known benchmark problem: flow in a lid-driven cavity. The dimensions domain in x-direction and y-direction are [0,1]. We used a mesh with 1563 nodes and 2988 elements and the following Reynolds numbers (Re): 10, 100, 400 and 1000. The results were compared with Ghia *et al.* (1982) and Marchi *et al.* (2009).

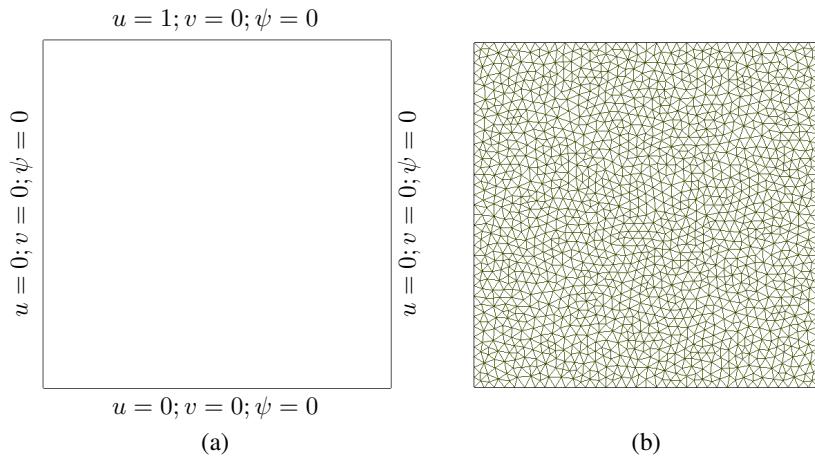


Figure 1: Lid-driven cavity: (a) geometry (b) mesh.

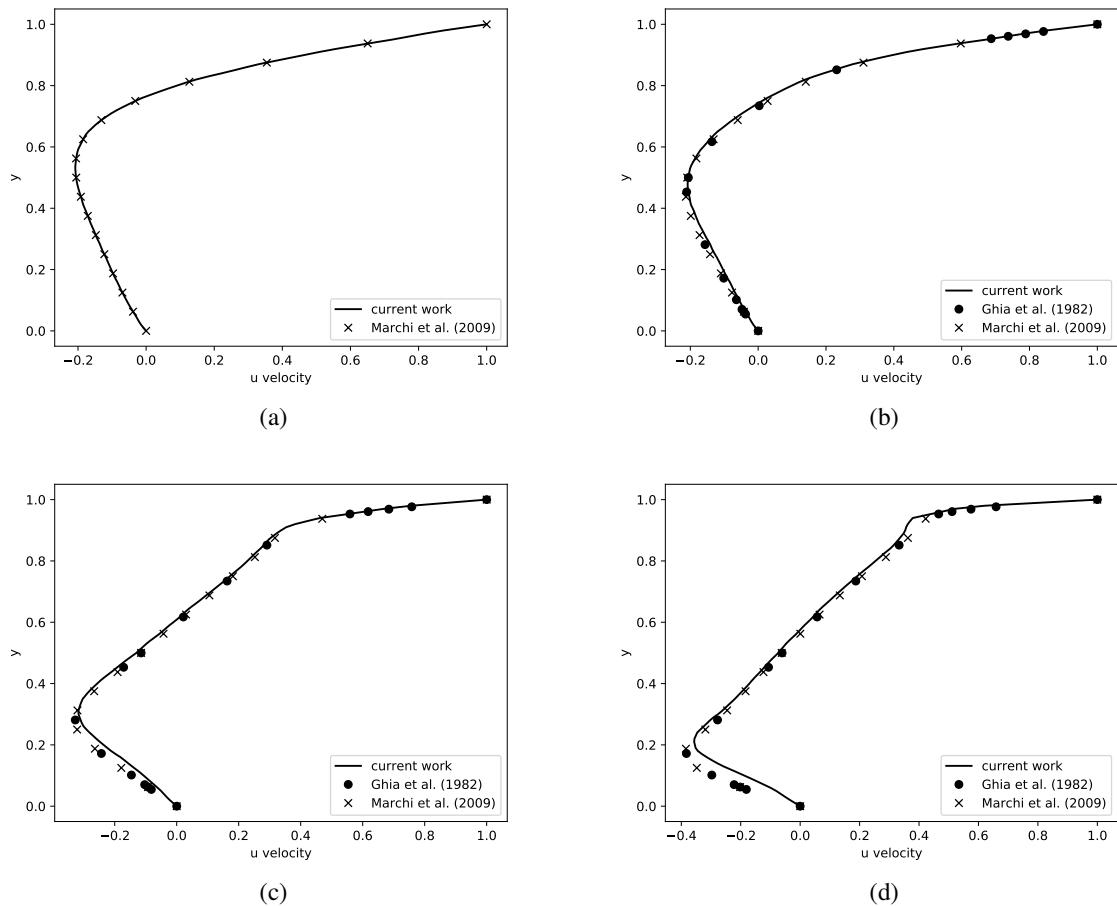


Figure 2: Centerline velocity profile ($x = 0.5$) in a lid-driven cavity for different Reynolds numbers: (a) 10 (b) 100 (d) 400 (f) 1000.

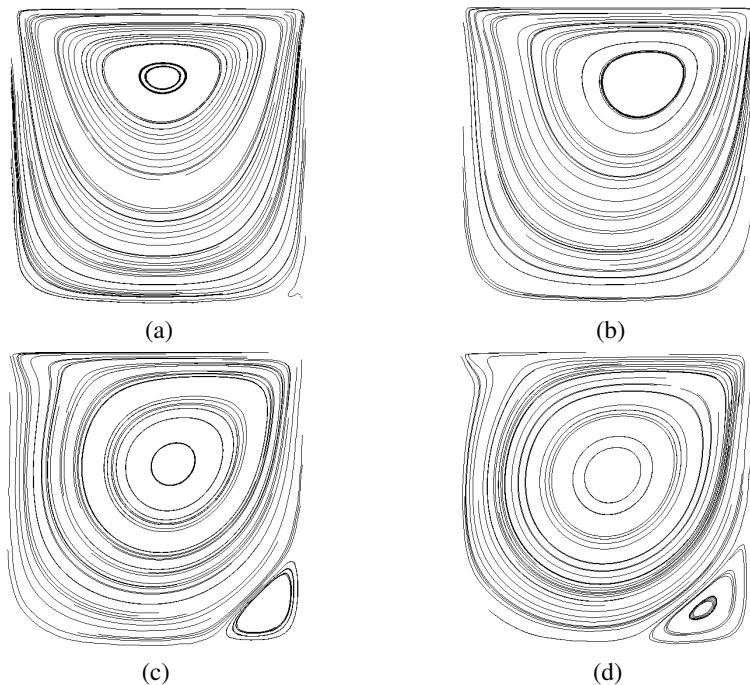


Figure 3: Streamfunction countours in a lid-driven cavity for different Reynolds numbers: (a) 10 (b) 100 (c) 400 (d) 1000.

4. RESULTS AND DISCUSSION

Some results of simulations are shown to demonstrate its capability of using unstructured triangular meshes on various geometries and combination of geometries. Numerical results are given for several cases of blood flows in artery. The post-processing was performed by open source software *PARAVIEW* proposed by Henderson (2007). The lumen diameter of a typical artery is about $D = 0.003\text{m}$, viscosity in the lumen are set to $\mu = 0.0035\text{Pa.s.}$ and density $\rho = 1060\text{kg/m}^3$ as suggested by Bozsak *et al.* (2014). According to Kessler *et al.* (1998), the velocity of the flow at coronary artery is $v = 12\text{cm/s.}$ Therefore, the Reynolds number is $Re = 109.$ Four different geometries were used in the simulations as proposed by Wang *et al.* (2017) and is shown in the Figure 4. Only half domain are shown since the results are symmetric in y-direction.

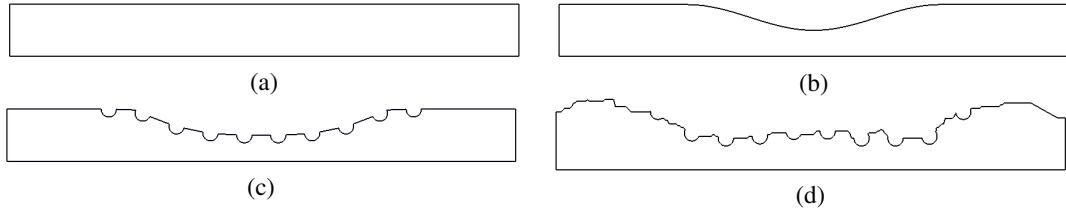


Figure 4: Non-dimensional geometry for blood flow dynamics in coronary arteries. The channel length $L = 10R$ is based on the channel width $R = 1.$ (a) Straight Channel (b) Curved Channel (c) Curved Channel with Stent and (d) Real Channel with Stent.

4.1 Straight Channel

For the case where we don't have atherosclerosis, the coronary artery is a straight channel. In the literature, this case is known as Hagen-Poiseuille flow and there is an analytical solution for the velocity profile. The analytical solution is given by:

$$u = u_{max} \left(1 - \frac{y^2}{R^2} \right) \quad (10)$$

where u_{max} is the velocity in the symmetry axis and has the value $u_{max} = 2,$ R is non-dimensional radius and has the value $R = 1$ and y the width of half vessel and ranges from $y = [0, 1].$

The Figure 5 shows the transient velocity profile along the y-direction at the middle of the channel ($x = 5.0R$). As expected, the velocity profile evolution in time converges to the analytical solution with a small error.

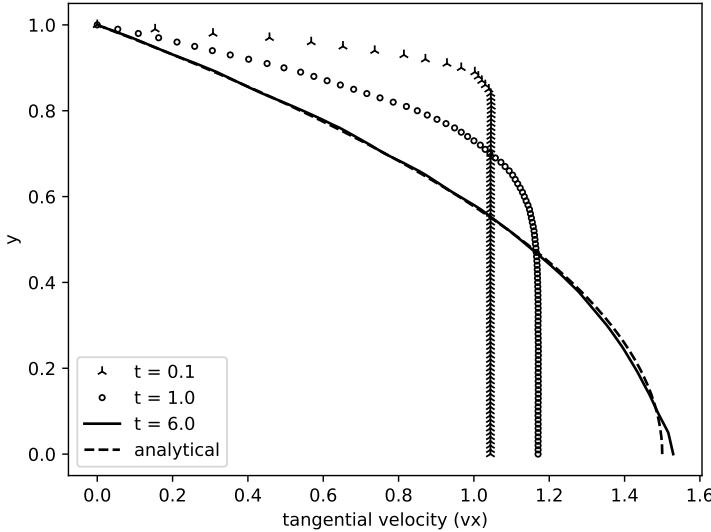


Figure 5: Velocity field profile evolution in time for Straight Channel.

The Figure ?? shows the velocity field evolution in space and time for half domain since the results are symmetric in y-direction. The velocity field is represented with non-dimensional values where the red color is $vx = 2$ and the blue color is $vx = 0.$

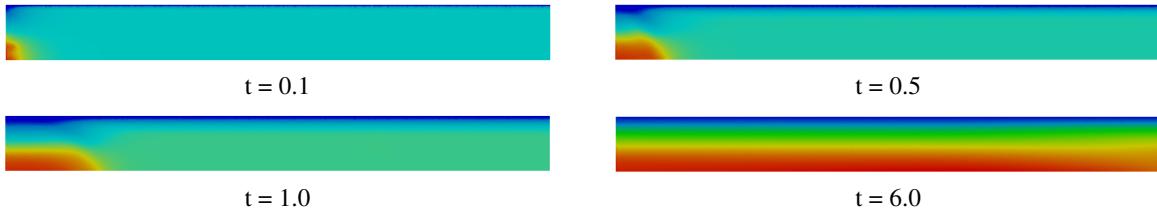


Figure 6: Time evolution of the velocity field for the straight geometry channel. The velocity profile at the middle of the channel is showed in Fig. 5.

4.2 Curved Channel

For the case where we have atherosclerosis, the coronary artery is a curved channel. This geometry promotes a smooth reduction of the channel diameter followed by a smooth expansion as commonly found in Venturi channel. Was considered 40% of obstruction, that is, the vessel has 60% of diameter for blood flow. The Figure 7 shows the transient velocity profile along the y-direction at the middle of the channel ($x = 5.0R$).

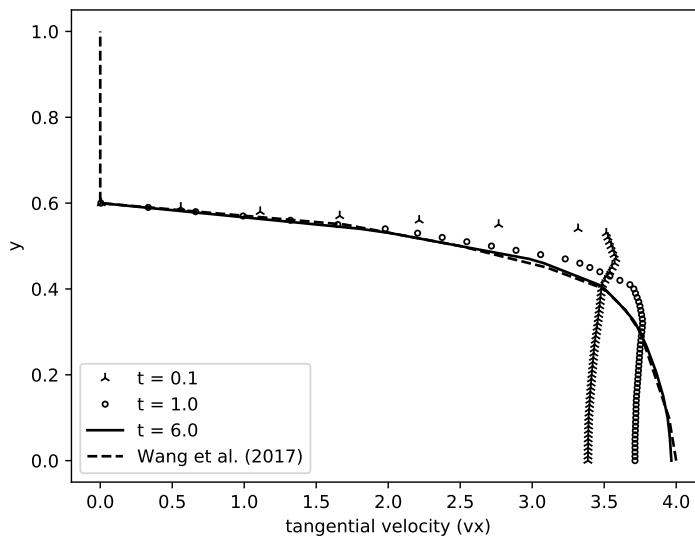


Figure 7: Velocity field profile evolution in time for Curved Channel.

The Figure 8 shows the velocity field evolution in space and time for half domain since the results are symmetric in y-direction. The velocity field is represented with non-dimensional values where the red color is $vx = 4$ and the blue color is $vx = 0$.

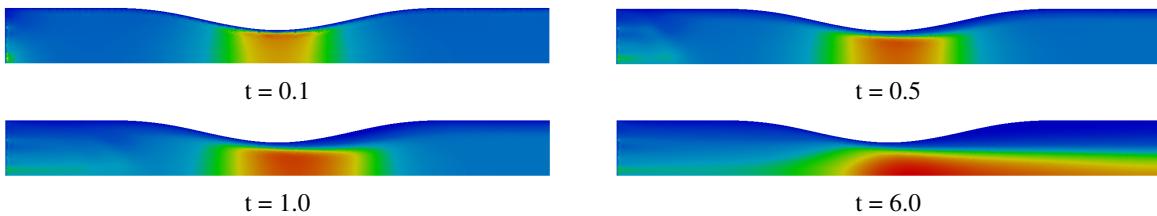


Figure 8: Time evolution of the velocity field for the curved geometry channel. The velocity profile at the middle of the channel is showed in Fig. 7.

4.3 Curved Channel with Stent

In this case, the stent strut was placed at the top of the curved channel and it was modeled by 10 semi-circles uniformly spaced. As in other cases was considered 40% of obstruction, that is, the vessel has 60% of diameter for blood flow. The Figure 9 shows the transient velocity profile along the y-direction at the middle of the channel ($x = 5.0R$).

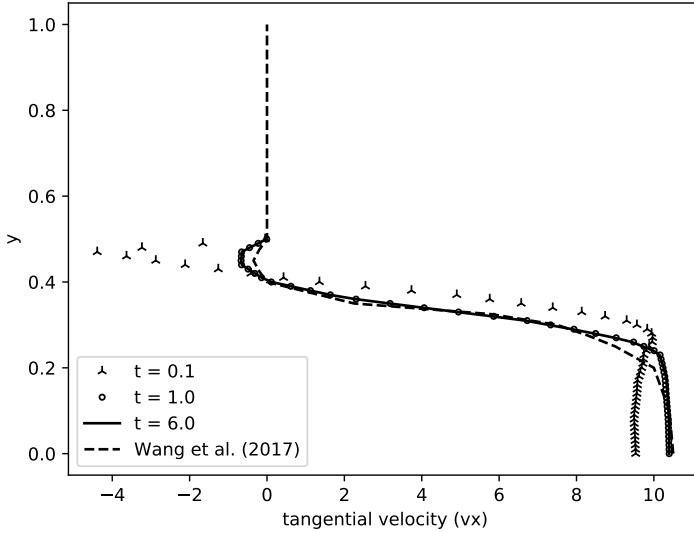


Figure 9: Velocity field profile evolution in time for Curved Channel with Stent.

The Figure 10 shows the velocity field evolution in space and time on the left hand side and the concentration field evolution on the right hand side for half domain since the results are symmetric in y-direction. The velocity field is represented with non-dimensional values where the red color is $vx = 10$ and the blue color is $vx = 0$. The concentration field also is represented with non-dimensional values where the red color is $c = 1$ and the blue color is $c = 0$.

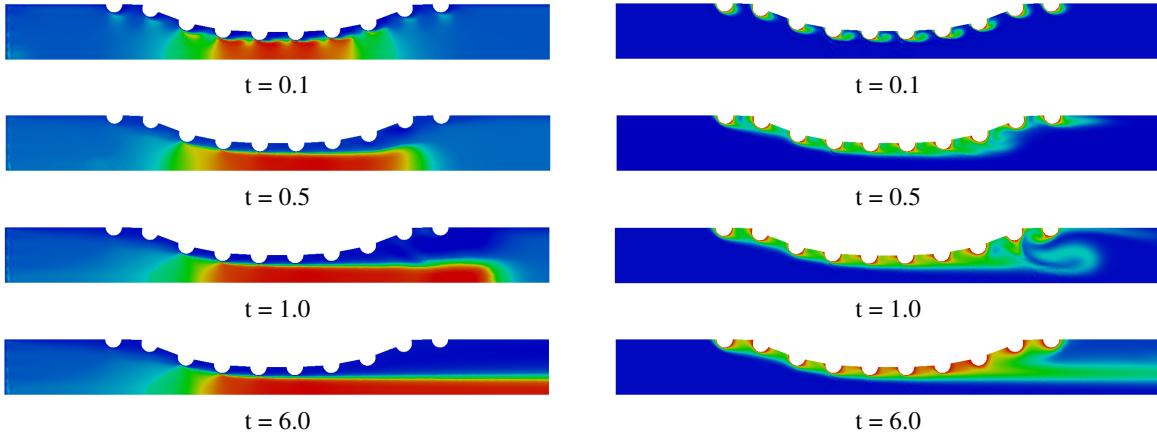


Figure 10: Time evolution of the velocity (left) and concentration (right) fields for the curved geometry channel with stent strut. The velocity profile at the middle of the channel is showed in Fig. 9.

4.4 Real Channel with Stent

In this case, a numerical simulation was performed for a real artery channel whose geometry was taken using image processing from a real coronary artery photography as proposed by Wang *et al.* (2017). It is important to know that each coronary artery geometry is particular to each patient and compatible to its health conditions. The stent strut was placed at the top of the curved channel and it was modeled by 10 semi-circles uniformly spaced. As in other cases was considered 40% of obstruction, that is, the vessel has 60% of diameter for blood flow. The Figure 11 shows the transient velocity profile along the y-direction at the middle of the channel ($x = 5.0R$).

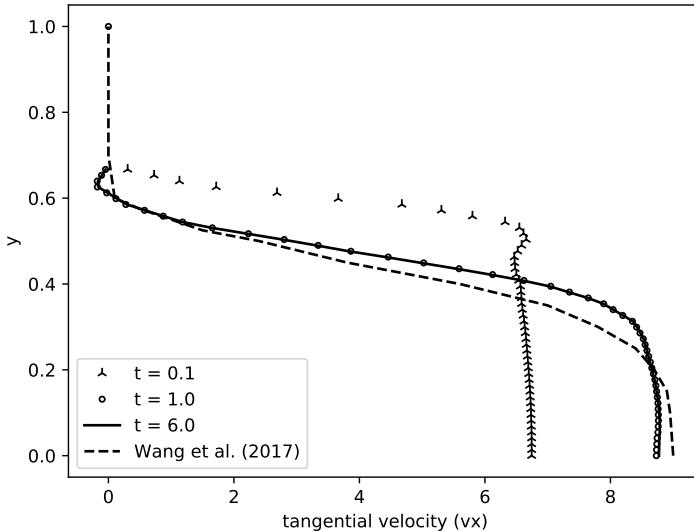


Figure 11: Velocity field profile evolution in time for Real Channel with Stent.

The Figure 12 shows the velocity field evolution in space and time on the left hand side and the concentration field evolution on the right hand side for half domain since the results are symmetric in y -direction. The velocity field is represented with non-dimensional values where the red color is $vx = 9$ and the blue color is $vx = 0$. The concentration field also is represented with non-dimensional values where the red color is $c = 1$ and the blue color is $c = 0$.

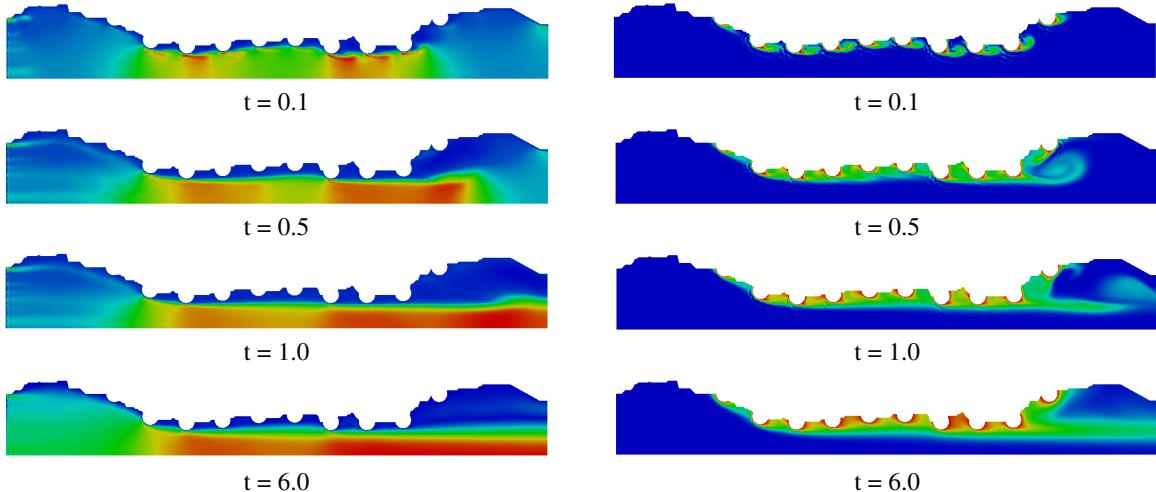


Figure 12: Time evolution of the velocity (left) and concentration (right) fields for the curved geometry channel with stent strut. The velocity profile at the middle of the channel is showed in Fig. 11.

5. CONCLUSION

The results obtained in this paper for the velocity and concentration fields of complex geometries of a modeled coronary artery channel were similar to those presented by Wang *et al.* (2017). However, the numerical simulation was performed using the streamfunction and vorticity formulation by finite element method approach. The Taylor-Galerkin Method was applied to the decrease the spurious oscillations as seen for moderate to high Reynolds numbers. The validation of the numerical solution was done by well-known benchmark lid-driven cavity problem and the results were compared with others authors as well as the Hagen-Poiseuille flow for the case straight channel that was compared with analytical solution. The streamfunction and vorticity formulation showed an useful approximation for to calculate the velocity and concentration fields since the variables are scalars allowing then a smooth implementation.

6. ACKNOWLEDGEMENTS

The authors thank the FAPERJ (Research Support Foundation of the State of Rio de Janeiro) for its financial support.

7. REFERENCES

- Anjos, G.R., 2012. *A 3D ALE Finite Element Method for Two-Phase Flows with Phase Change*. Ph.D. thesis, École Polytechnique Fédérale de Lausanne.
- Bozsak, F., J-M., C. and Barakat, A., 2014. "Modeling the transport of drugs eluted from stents: physical phenomena driving drug distribution in the arterial wall". *Biomech Model Mechanobiol*, Vol. 13, pp. 327–347. doi:10.1007/s10237-013-0546-4.
- Donea, J., 1984. "A taylor-galerkin method for convective transporte problems". *International Journal for Numerical Methods in Engineering*.
- Geuzaine, C. and Remacle, J., 2009. "Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities". *International Journal for Numerical Methods in Engineering*.
- Ghia, U., Ghia, K.N. and Shin, C.T., 1982. "High-re solutions for incompressible flow using the navier-stokesequations and a multi-grid." *Journal of Computational Physics*.
- Henderson, A., 2007. "Paraview guide, a parallel visualization application". *Kitware Inc*.
- Jones, E., Oliphant, T. and Peterson, P., 2001. "Scipy - open source scientific tools for python". URL <http://www.scipy.org/>. [Online; accessed 15/03/2018 09:14].
- Kessler, W., Moshage, W., Galland, A., Zink, D., Achenbach, S., Nitz, W., Laub, G. and Bachmann, K., 1998. "Assessment of coronary blood flow in humans using phase difference mr imaging comparison with intracoronary doppler flow measurement." *International Journal of Cardiac Imaging*.
- Lohner, R., Morgan, K. and Zienkiewicz, O.C., 1984. "The solution of non-linear hyperbolic equation systems by the finite element method". *International Journal of Numerical Methods in Fluids*.
- Marchi, C.H., Suero, R. and Araki, L.K., 2009. "The lid-driven square cavity flow: Numerical solution with a 1024 x 1024 grid." *Journal of the Brazilian Society of Mechanical Sciences and Engineering*.
- McGinty, S. and Pontrelli, G., 2016. "On the role of specific drug binding in modelling arterial eluting stents". *Journal of Mathematical Chemistry*.
- Python, S.F., 1990. "Python language reference, version 2.7". URL <http://www.python.org/>. [Online; accessed 12/12/2017 12:18].
- Wang, H., McGinty, S., Lucena, R., Pontes, J., Anjos, G. and Mangiavacchi, N., 2017. "Dynamics of blood flow in coronary artery". *International Congress of Mechanical Engineering*.
- World Health Organization, W., 2017. "Cardiovascular diseases". URL <http://www.who.int/mediacentre/factsheets/fs317/en/>. [Online; accessed 28/03/2018 14:37].
- Zienkiewicz, O.C. and Taylor, R.L., 2000. "Finite element method - volume 3: Fluid dynamics". *Butterworth-Heinemann*.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.

BLOOD FLOW SIMULATION USING STREAM FUNCTION-VORTICITY FEM FORMULATION

L. MARQUES¹, G. R. ANJOS² AND J. PONTES³

Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia Mecânica

¹ marquesleandro67@gmail.com, ² gustavo.anjos@uerj.br and ³ jose.pontes@uerj.br

1. ABSTRACT

The present work aims at developing a computational framework to simulate coronary artery flows in cartesian coordinates. An accurate method capable of capturing the flow dynamics is strictly required. In this paper a Finite Element Method (FEM) is used to solve the governing equations of the motion of the blood flow found in coronary artery as incompressible fluid using the stream-vorticity formulation with coupled species transport equation.

Keywords: Finite Element Method, Taylor-Galerkin Method, Coronary Artery, Blood.

2. INTRODUCTION

According to the *Health World Organization*, more people die annually from the cardiovascular diseases (CVDs) than from any other cause in the world. The leading cause of these deaths is coronary artery with atherosclerosis and the main purpose of stent implementation is to attack this problem. This work aims to know how the dynamics of blood flow in coronary artery with atherosclerosis and with stents struts placed.

We consider the hydrodynamic of the blood flow in the artery in the vicinity of a stent strut for a two-dimensional domain. The difference of artery shapes and existence of the struts have been investigated in 4 test cases in the influence on the flow dynamics and the transport of chemical species as suggested by Wang et al. (2017).

We employ the Finite Element method on an unstructured mesh created by open source software called *GMSH* for the discretization of the incompressible single-phase Navier-Stokes through the stream-vorticity function coupled with species transport equation. We apply the Taylor-Galerkin method to the decrease the spurious oscillations as seen for moderate to high Reynolds numbers as proposed by Donea (1984).

3. MATHEMATICAL MODEL

A two-dimensional Finite Element Method approach is employed to analyse the dynamics of blood flow in coronary artery and possible influence of stents struts. The modelling of the problem can be described by a formulation using the vorticity and stream function applied in the conservation of momentum equation for incompressible flow and the species transport equation.

$$\frac{\partial \omega}{\partial t} + \mathbf{v} \cdot \nabla \omega = \frac{1}{Re} \nabla^2 \omega \quad (1)$$

$$\nabla^2 \psi = -\omega \quad (2)$$

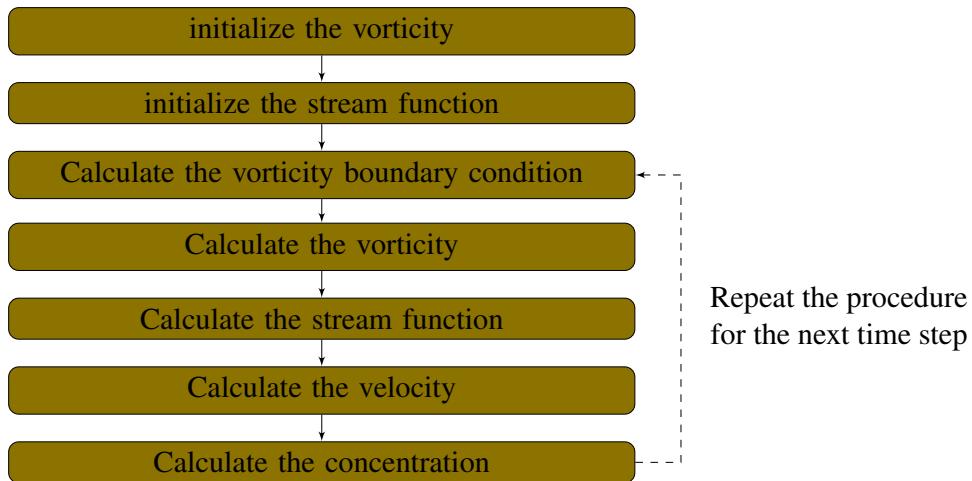
$$\mathbf{v} = \mathbf{D}\psi \quad (3)$$

$$\frac{\partial c}{\partial t} + \mathbf{v} \cdot \nabla c = \frac{1}{ReSc} \nabla^2 c \quad (4)$$

Where ω is the vorticity field, ψ is the stream function field, c is the concentration scalar field of the chemical species, $\mathbf{v} = (v_x, v_y)$ is the velocity field, $\mathbf{D} = [\partial/\partial y, -\partial/\partial x]$ is a mathematical operator, $Re = \rho u D / \mu$ is the Reynolds number, $Sc = \nu/D$ is the Schmidt number, x and y are the independent spatial variables and t is the time variable.

The temporal domain was discretized using the Taylor series with the terms higher second order omitted. The second order terms were used to decrease the spurious oscillations for moderate to high Reynolds numbers as proposed by Donea (1984). The spatial domain was discretized by Galerkin Formulation using linear triangular.

The linear system of equations that come from implementing the FEM is solved through an iterative method *Conjugate Gradient Solver* available in the public library for scientific tools *SciPy* in the *Python language*. The solution algorithm used is shown below:



4. RESULTS

Numerical results are shown for several cases of blood flows in artery. The post-processing was performed by open source software *PARAVIEW*. The lumen radius of a typical artery is about $R = 0.0015\text{m}$, viscosity in the lumen are set to $\mu = 0.0035\text{Pa.s.}$ and density $\rho = 1060\text{kg/m}^3$ as suggested by Bozsak et al. (2014). According to Kessler et al. (1998), the velocity of the flow at coronary artery is $v = 12\text{cm/s.}$ Therefore, the Reynolds number is $Re = 54.5.$ Four non-dimensional geometries were used with channel length $L = 10R$ and channel width $R = 1$ as proposed by Wang et al. (2017), however were modified to cartesian coordinates and

are shown in the Figure 1. Only half domain are shown since the results are symmetric in y -direction.

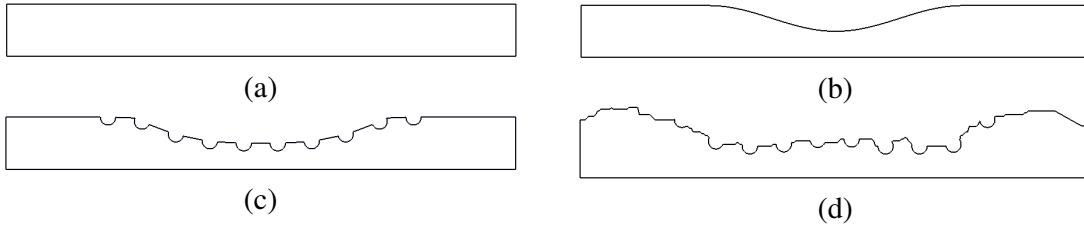


Figura 1 – Non-dimensional geometry for blood flow dynamics in coronary arteries. (a) Straight Channel (b) Curved Channel (c) Curved Channel with Stent and (d) Real Channel with Stent.

In fig 2 and 3, are shown the velocity profiles evolution in time and the velocity and concentration fields for the steady state respectively. There is a significant difference between the results shown and those shown in axisymmetric coordinates by Wang et al. (2017).

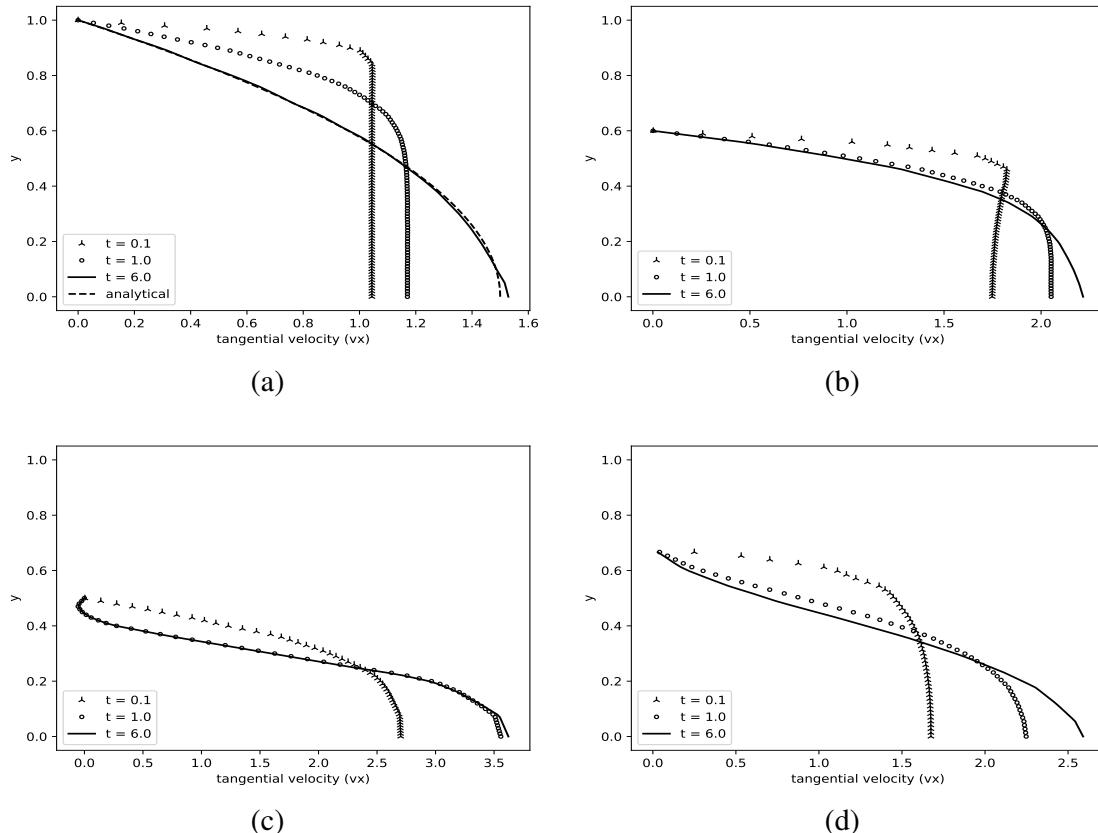


Figura 2 – Velocity field profile evolution in time for various geometries (a) Straight Channel (b) Curved Channel (c) Curved Channel with Stent and (d) Real Channel with Stent.

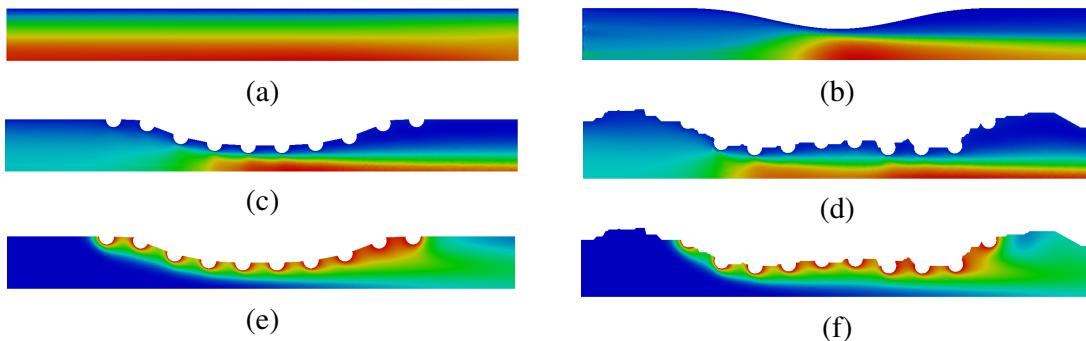


Figura 3 – Velocity field steady state: (a) Straight Channel (b) Curved Channel (c) Curved Channel with Stent and (d) Real Channel with Stent. Concentration field steady state: (e) Curved Channel with Stent and (f) Real Channel with Stent.

5. CONCLUSION

In this paper, the results were shown for two-dimensional domain in complex geometries of modeled coronary artery channel. The numerical simulation was performed using the streamfunction and vorticity formulation with coupled species transport equation by finite element method approach. The streamfunction and vorticity formulation showed a smooth implementation for calculating the variables since they are scalars. However, there is a significant difference between the results shown in cartesian coordinates and those shown by Wang et al. (2017) in axisymmetric coordinates.

6. REFERENCES

- BOZSAK, F.; J-M., C.; BARAKAT, A. Modeling the transport of drugs eluted from stents: physical phenomena driving drug distribution in the arterial wall. *Biomech Model Mechanobiol*, 13, 327–347, 2014.
- DONEA, J. A taylor-galerkin method for convective transport problems. *International Journal for Numerical Methods in Engineering*, 1984.
- KESSLER, W.; MOSHAGE, W.; GALLAND, A.; ZINK, D.; ACHENBACH, S.; NITZ, W.; LAUB, G.; BACHMANN, K. Assessment of coronary blood flow in humans using phase difference mr imaging comparison with intracoronary doppler flow measurement. *International Journal of Cardiac Imaging*, 1998.
- WANG, H.; MCGINTY, S.; LUCENA, R.; PONTES, J.; ANJOS, G.; MANGIAVACCHI, N. Dynamics of blood flow in coronary artery. *International Congress of Mechanical Engineering*, 2017.

