



**Universidade do Estado do Rio de Janeiro**  
Centro de Ciência e Tecnologia  
Faculdade de Engenharia

Hugo Checo Silva

**Finite elements in convection dominated flows: a semi-Lagrangian method**

Rio de Janeiro  
2011

Hugo Checo Silva

**Finite elements in convection dominated flows: a  
semi-Lagrangian method**



Advisor: Prof. Dr. Norberto Mangiavacchi

Rio de Janeiro  
2011

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

C514 Checo Silva, Hugo  
Elementos finitos em fluidos dominados pelo fenômeno de advecção: um método semi-Lagrangeano / Hugo Checo Silva.  
– 2011.  
69f.

Orientador: Norberto Mangiavacchi.  
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Método dos elementos finitos - Teses. 2. Engenharia Mecânica. I. Mangiavacchi, Norberto. II. Universidade do Estado do Rio de Janeiro. III. Título.

CDU 621:519.62

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

---

Assinatura

---

Data

Hugo Checo Silva

## **Finite elements in convection dominated flows: a semi-Lagrangian method**

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

Aprovado em: dia 7 de julho de 2011.

Banca Examinadora:

---

Prof. Dr. Norberto Mangiacacchi (Orientador)  
Faculdade de Engenharia Mecânica - UERJ

---

Prof. Dr. Luiz Mariano Carvalho  
Instituto de Matemática e Estatística - UERJ

---

Prof. Dr. Carlos Antônio de Moura  
Instituto de Matemática e Estatística - UERJ

---

Prof. Dr. José da Rocha Miranda Pontes  
Universidade Federal do Rio de Janeiro - UFRJ

---

Prof. Dr. Luis Manuel Portela  
Delft University of Technology - Países Baixos

## **ACKNOWLEDGMENTS**

I wish to thank in first place my advisor, Professor Norberto Mangiavacchi. Everything was made possible with his help during this two years: this work is the culmination of that process, and hopefully the beginning of a new one.

Special thanks to professor Christian Schaefer, who showed me the way into getting a masters degree at the UERJ.

Thanks to CAPES for the financial support.

I have also to thank a lot of friends for their support, they directly and indirectly influenced this work. First of all to three of them: Leon Matos, Hyun Ho Shin and Pedro Torres for receiving me at their home in Rio de Janeiro and helping me in many different ways. This work was made easier by them. They thought me much in different subjects relative to finite elements; especial regards to Pedro for his invaluable instructions on networks, Linux-based systems and parallelism. I'm also grateful to my partner in the masters degree courses, Manolo Solalinde.

I cannot forget my coworker, friend, and personal portuguese teacher (slang included!) Raama Costa. The hours spent together (programming!) will always be remembered.

Thanks to my dear friend Laura Arévalos, for your moral support and wisdom: you helped me pass through very difficult times, I will always be thankful.

To our little FIUNA community in Rio de Janeiro, true friends from old times: Carlos Sauer, Carlos Galeano and Mauricio Poletti.

Many thanks to the GESAR (development group at the UERJ) staff: Sonia and Jorge, professors Carlos Moura, Luiz Mariano Carvalho, Francisco Soeiro, Rogerio Saldanha da Gama, and to my laboratory partners: André, Cristiane, Eduardo, Gustavo, Igor and Michele.

Very special thanks to my family: Carla, Cheche, mother and father.

## RESUMO

CHECO SILVA, Hugo. *Finite elements in convection dominated flows: a semi-Lagrangean method.* Brasil. 2011. 69 f. Dissertação (Mestrado em Engenharia Mecânica) – Faculdade de Engenharia Mecânica, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2011.

Os escoamentos altamente convectivos representam um desafio na simulação pelo método de elementos finitos. Com a solução de elementos finitos de Galerkin para escoamentos incompressíveis, a matriz associada ao termo convectivo é não simétrica, e portanto, a propriedade de aproximação ótima é perdida. Na prática as soluções apresentam oscilações espúrias. Muitos métodos foram desenvolvidos com o fim de resolver esse problema. Neste trabalho apresentamos um método semi-Lagrangeano, o qual é implicitamente um método do tipo *upwind*, que portanto resolve o problema anterior, e comparamos o desempenho do método na solução das equações de convecção-difusão e Navier-Stokes incompressível com o Streamline Upwind Petrov Galerkin (SUPG), um método estabilizador de reconhecido desempenho. No SUPG, as funções de forma e de teste são tomadas em espaços diferentes, criando um efeito tal que as oscilações espúrias são drasticamente atenuadas. O método semi-Lagrangeano é um método de fator de integração, no qual o fator é um operador de convecção que se desloca para um sistema de coordenadas móveis no fluido, mas restabelece o sistema de coordenadas Lagrangeanas depois de cada passo de tempo. Isto prevê estabilidade e a possibilidade de utilizar passos de tempo maiores. Existem muitos trabalhos na literatura analisando métodos estabilizadores, mas não assim com o método semi-Lagrangeano, o que representa a contribuição principal deste trabalho: reconhecer as virtudes e as fraquezas do método semi-Lagrangeano em escoamentos dominados pelo fenômeno de convecção.

Palavras-chave: Semi-lagrangiano. SUPG (Streamline Upwind Petrov Galerkin). FEM (Método dos elementos finitos). Método estabilizado.

## ABSTRACT

Convection dominated flows represent a challenge for finite element method simulation. Many methods have been developed to address this problem. In this work we compare the performance of two methods in the solution of the convection-diffusion and Navier-Stokes equations on environmental flow problems: the Streamline Upwind Petrov Galerkin (SUPG) and the semi-Lagrangian method. In Galerkin finite element methods for fluid flows, the matrix associated with the convective term is non-symmetric, and as a result, the best approximation property is lost. In practice, solutions are often corrupted by spurious oscillations. In this work, we present a semi-Lagrangian method, which is implicitly an *upwind* method, therefore solving the spurious oscillations problem, and a comparison between this semi-Lagrangian method and the Streamline Upwind Petrov Galerkin (SUPG), an stabilizing method of recognized performance. The SUPG method takes the interpolation and the weighting functions in different spaces, creating an effect so that the spurious oscillations are drastically attenuated. The semi-Lagrangian method is a integration factor method, in which the factor is an operator that shifts to a coordinate system that moves with the fluid, but it resets the Lagrangian coordinate system after each time step. This provides stability and the possibility to take bigger time steps. There are many works in the literature analyzing stabilized methods, but they do not analyze the semi-Lagrangian method, which represents the main contribution of this work: to recognize the strengths and weaknesses of the semi-Lagrangian method in convection dominated flows.

Keywords: Semi-lagrangian. SUPG (Streamline Upwind Petrov Galerkin). FEM (Finite element method). Stabilized method.

## LIST OF FIGURES

Figure 1 - Mesh generation with FERSim . . . . .	13
Figure 2 - Some results with FERSim . . . . .	13
Figure 3 - Some LBB-stable elements: circles represent velocity nodes, and triangles pressure nodes . . . . .	34
Figure 4 - Problem <i>characteristics</i> . . . . .	36
Figure 5 - Interpolation in the semi-Lagrangian method . . . . .	37
Figure 6 - Rotating cone problem statement . . . . .	45
Figure 7 - Results after a full rotation: (a) Galerkin; (b) SUPG (c) semi-Lagrangian	46
Figure 8 - Results after a full rotation: (a) Galerkin; (b) SUPG (c) semi-Lagrangian	47
Figure 9 - Results on the cone impinging a mesh boundary (4 node quadrilaterals instead of 9). (a): semi-Lagrangian; (b) SUPG. . . . .	48
Figure 10 - semi-Lagrangian results with 4 node quadrilaterals and a finer mesh . .	48
Figure 11 - Lid-driven cavity problem statement . . . . .	49
Figure 12 - Lid driven cavity 2D meshes: (a) 2.000 elements. (b) 5.000 elements. .	50
Figure 13 - Lid driven cavity 2D meshes: (a) 10.000 elements. (b) 40.000 elements. .	50
Figure 14 - Position of vortex until Re=10.000 . . . . .	51
Figure 15 - Velocity profiles at $x = 0.5$ (left) and $y = 0.5$ (right) (Re = 1.000). (a) and (b): mesh of 40.000 elements; (c) and (d): mesh of 2.000 elements. .	52
Figure 16 - Velocity profiles at $x = 0.5$ (left) and $y = 0.5$ (right) (Re = 5.000). (a) and (b): mesh of 40.000 elements; (c) and (d): mesh of 2.000 elements. .	53
Figure 17 - Velocity profiles at $x = 0.5$ (left) and $y = 0.5$ (right) (Re = 10.000). (a) and (b): mesh of 40.000 elements; (c) and (d): mesh of 2.000 elements. .	54
Figure 18 - SUPG streamlines for: (a) Re = 1.000; (b) Re = 5.000; (c) Re = 10.000;	55
Figure 19 - Semi-Lagrangian streamlines for Re=1.000. CFL: (a) 1.1375; (b) 1.52; (c) 2.275; (c) 4.55. . . . .	55
Figure 20 - Semi-Lagrangian streamlines for Re=5.000. CFL: (a) 1.1375; (b) 1.52; (c) 2.275; (c) 4.55. . . . .	56
Figure 21 - Semi-Lagrangian streamlines for Re=10.000. CFL: (a) 1.1375; (b) 1.52; (c) 2.275; (c) 4.55. . . . .	57
Figure 22 - Error: (a) Re=1.000; (b) Re=5.000; (c) Re=10.000; . . . . .	59
Figure 23 - Results for Reynolds=20.000: (a) semi-Lagrangian; (b) SUPG. . . . .	59
Figure 24 - Backward facing step problem statement . . . . .	60
Figure 25 - Backward facing step mesh . . . . .	61
Figure 26 - Semi-Lagrangian (left column) and SUPG results (right columns) for $Re=800$ , $Re=1200$ and $Re=1600$ (top, middle and bottom line respectively). . . . .	62

Figure 27 - (a) SUPG and (b) semi-Lagrangian results for an 16.000 elements mesh  
and  $\Delta t = 0.1$  . . . . . 63

Figure 28 - semi-Lagrangian with linear interpolation ( $Re = 3.000$ ): (a) 16.000 ele-  
ments mesh; (b) 120.000 elements mesh. . . . . 64

## **LIST OF TABLES**

Table 1 - Number of resolved vortices in function of Reynolds numbers . . . . .	54
Table 2 - Vortex coordinates . . . . .	58
Table 3 - Compared execution times . . . . .	64

# CONTENTS

<b>INTRODUCTION</b>	11
<b>1 GOVERNING EQUATIONS</b>	15
1.1 Continuity (conservation of mass) equation and incompressibility constraint	15
1.2 The Navier-Stokes equations	15
1.2.1 <u>Adimensionalization</u>	17
1.3 The convection-diffusion equation	17
1.3.1 <u>Adimensionalization</u>	18
1.4 Boundary and initial conditions	18
<b>2 DISCRETIZATION METHOD AND SUPG FORMULATION</b>	21
2.1 Variational formulation	21
2.2 Streamline Upwind/ Petrov Galerkin method	23
2.2.1 <u>The <math>\tau_{SUPG}</math> stabilization parameter</u>	24
2.3 Discretization and solution method	25
2.3.1 <u>Spatial discretization</u>	25
2.3.2 <u>Newton-Krylov solution method</u>	26
2.3.3 <u>Matrix form of the equations</u>	30
2.3.4 <u>Discretization in time</u>	32
2.3.5 <u>Mesh elements</u>	34
<b>3 SEMI-LAGRANGIAN METHOD</b>	35
3.1 Introduction	35
3.2 Variational formulation	36
3.3 Discretization and solution method	37
3.3.1 <u>Spatial discretization</u>	37
3.3.2 <u>Matrix form of the equations</u>	40
3.3.3 <u>Discretization in time</u>	42
<b>4 NUMERICAL TESTS</b>	43
4.1 Introduction	43

4.2	<b>The libMesh library</b>	43
4.3	<b>Tests on the Convection-Diffusion equation</b>	44
4.3.1	<u>The rotating cone test</u>	44
4.3.2	<u>Rotating cone test results</u>	45
4.3.3	<u>The cone impinging a mesh boundary test</u>	46
4.3.4	<u>Cone impinging a mesh boundary test results</u>	47
4.4	<b>Tests on the Navier-Stokes equations</b>	48
4.4.1	<u>The 2D lid-driven cavity test</u>	48
4.4.2	<u>2D lid-driven cavity test</u>	51
4.4.3	<u>The backward facing step test</u>	60
4.4.4	<u>Backward facing step test results</u>	61
4.4.5	<u>Execution times</u>	64
5	<b>SUMMARY</b>	65
5.1	<b>Future Research Area</b>	65
	<b>REFERENCES</b>	66
	<b>APPENDIX A – Libmesh.implementation</b>	70

## INTRODUCTION

The basis of fluid dynamics models are the Navier-Stokes (N-S) equations, a set of nonlinear partial differential equations. Transport phenomena are described using the convection-diffusion equation along with the N-S equations. There are many physical situations where fluid is approximately incompressible, and thus we have the incompressible N-S equations. The incompressible N-S and the convection-diffusion equations are used to model a large number of important phenomena, and that is the reason of the considerable efforts made in obtaining accurate and fast methods to solve it. They represent also a big challenge: the incompressibility restriction creates a coupling between pressure and velocity that makes the problem harder to solve, besides the nonlinearity issue.

The classical Galerkin approach has several drawbacks that have been identified: it may fail because of two reasons. First, in convection dominated flows, for which layers appear where the velocity solution and its gradient exhibit rapid variation, Galerkin leads to numerical oscillations in these layer regions which can spread quickly and degrade the entire solution domain (1). Stabilizing methods were originally developed to overcome this (2). Originally, T.J.R. Hughes and coworkers (2) developed methods that involve a new added term (the so called **stabilization term**), which is of the form of an operator multiplied by the residual of the differential equation; the operator defines the characteristics of the stabilization, and so we have different methods: SUPG (Streamline Upwind/Petrov Galerkin), GLS (Galerkin Least Squares), and the SUPG/PSPG (Pressure Stabilizing/ Petrov Galerkin) (3) developed by Tezduyar. Other stabilizing techniques have been developed in recent years: Finite Increment Calculus (Oñate, 1998) and variational multiscale models (Hughes, 1995), among others (4).

Second, the use of inappropriate combinations of interpolation functions to represent the velocity and pressure fields yields unstable schemes (5). This second problem can be overcome by means of taking the velocity interpolation functions of a higher degree than pressure interpolation functions (4).

GESAR (Grupo de Estudos e Simulações em Enchimento de Reservatórios - Research group on reservoir filling) is a development group that joins different areas of science such as mechanical engineering for fluid flow analysis, mathematics for resolution of the discretized equations of fluid flow, computational science for computational resource to simulate the fluid flow, geography for geographic information system (GIS) data analysis, biology and chemistry for biomass decomposition and water quality analysis. The objective of the group was to provide numerical simulation tools for 3D environmental flows. The result of their work is called FERSim (Finite Element Reservoir Simulator), a software written in C++. The most important classes are described

by Mangiavacchi et al. (2005), Dongala et al. (2006). The class *Simulator*, originally described by Anjos et al. (2007), is the hydrodynamic and scalar transport simulator.

In FERSim the method implemented to solve the Navier-Stokes and the scalar transport equations is a semi-Lagrangian one. This semi-Lagrangean method eliminates the non-linearity of this equations by means of considering the material derivative in the momentum equation, which has the following major advantages: being implicitly an upwind method, generating a symmetric system of equations, and giving the possibility to take bigger time steps (6). To solve the resulting system of equations on the Navier-Stokes equations, a projection method (velocity and pressure decoupled) is used. Linear interpolation is used to compute the velocity approximation for the semi-Lagrangian. For spatial discretization it makes use of the MINI element, a tetrahedral with a node in its centroid (or a triangle with a node in its centroid in the bidimensional case).

In this work we study the strengths and weaknesses of this method. As it is an upwind method, it is expected to diminish the numerical oscillations of the classical Galerkin approach. A comparison with a very well known stabilizing method, the Streamline Upwind Petrov Galerkin is made. SUPG takes the interpolation and the weighting functions in different spaces, creating an upwind effect, so that the spurious oscillations are drastically attenuated (2). Numerical experiments on benchmark problems were ran to make that comparison.

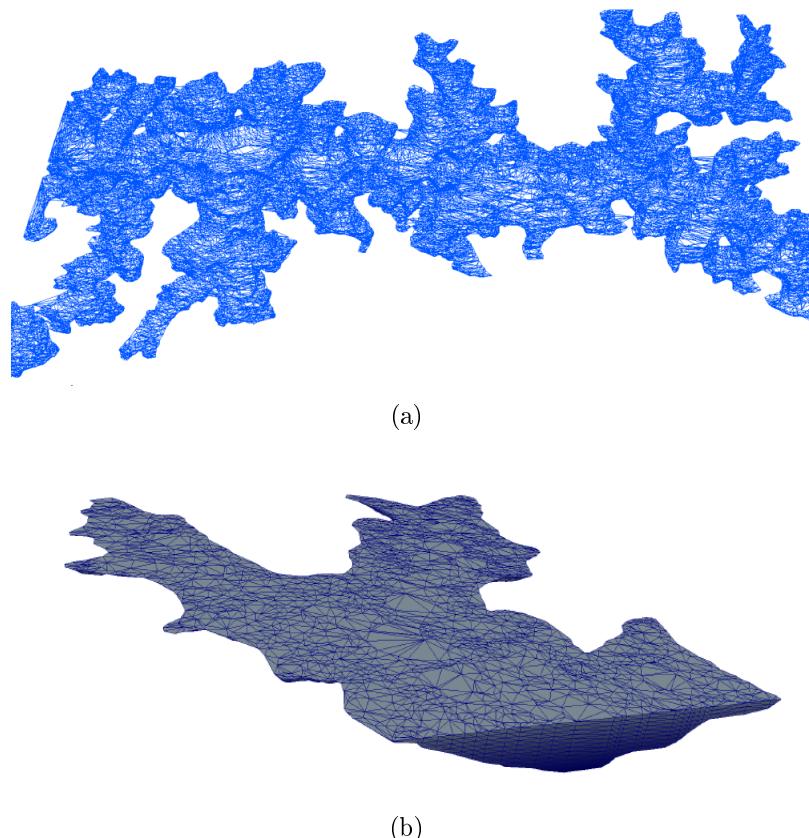


Figure 1 – Mesh generation with FERSim

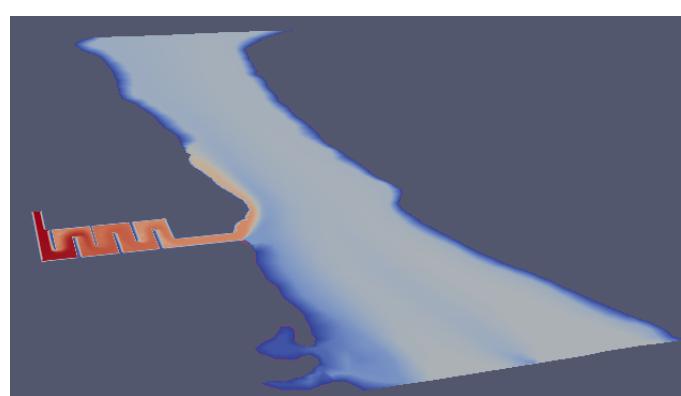


Figure 2 – Some results with FERSim

## Objectives

As mentioned above, the main objective of this work is to analize our semi-Lagrangean method for solving and stabilizing the classical Galerkin solution of the N-S and convection-diffusion equations.

Specific objectives are:

- To describe the governing equations of incompressible fluid flow and scalar transport (chapter 1).
- To describe the spatial Galerkin discretization and the SUPG formulation (chapter 2).
- To describe the semi-Lagrangian method implemented in this work (chapter 3).
- To perform numerical tests on benchmark problems (chapter 4).
- To present the computational results (chapter 4).
- To present the numerical tests analysis and draw conclusions (chapter 4).

## 1 GOVERNING EQUATIONS

In this chapter we will present a boundary value problem: the mass conservation, the Navier-Stokes and the scalar transport equations, along with the boundary and initial conditions.

### 1.1 Continuity (conservation of mass) equation and incompressibility constraint

If  $\rho$  is the fluid density,  $t$  is the time variable and  $\mathbf{v}$  is the fluid velocity, then the continuity equation in a cartesian coordinate system is:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} = 0 \quad (1.1)$$

Considering incompressibility, i.e., changes of density within the domain are negligible(the so called Boussinesq approximation, that states that the density differences at low Mach numbers - defined as  $Mc = \frac{U}{c}$ , where  $U$  is a characteristic velocity of the flow and  $c$  is the sound velocity for the same fluid - are sufficiently small and can be neglected (7)), then  $\frac{\partial \rho}{\partial t} = 0$ ,  $\nabla \rho = 0$ . The term  $\nabla \cdot \rho \mathbf{v}$  in equation 1.1 can be expanded as:

$$\nabla \cdot \rho \mathbf{v} = \nabla \rho \cdot \mathbf{v} + \rho \nabla \cdot \mathbf{v} \quad (1.2)$$

Substituting  $\frac{\partial \rho}{\partial t} = 0$ ,  $\nabla \rho = 0$  and 1.2 in 1.1 we obtain:

$$\nabla \cdot \mathbf{v} = 0 \quad (1.3)$$

which is the continuity equation for an incompressible fluid (5).

### 1.2 The Navier-Stokes equations

The momentum equations (motion equations also) are relations equating the rate of change of momentum of a selected portion of fluid and the sum of all forces acting in that portion of fluid (actually, we are considering just two forces: body forces - in the form of gravitational attraction - and surface forces). We have (4):

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = \nabla \cdot \sigma + \rho \mathbf{g} \quad (1.4)$$

where  $\mathbf{v}$  is the fluid velocity,  $\sigma$  is the stress tensor and  $\mathbf{g}$  is the gravity acceleration.

tion vector. It is convenient to split the stress tensor in the sum of two tensors: one due to pressure  $p$  and the other one  $\tau$  due to viscosity:

$$\sigma = -p\mathbf{I} + \tau \quad (1.5)$$

where  $p$  is the fluid pressure, and  $I$  is the identity tensor. The conservation of angular momentum shows that the stress tensor is symmetric. The term  $p\mathbf{I}$  is symmetric, so  $\tau$  must be too (4). For a Newtonian fluid, it is assumed that the stress tensor and the strain rate tensor ( $\tau$ ) are linearly related. The stress-strain relationship is given by (4):

$$\sigma = -p\mathbf{I} + \mu(\nabla\mathbf{v} + \nabla\mathbf{v}^T) + \lambda\nabla.\mathbf{v}\mathbf{I} \quad (1.6)$$

where  $\mu$  is the fluid dynamic viscosity and  $\lambda$  the so-called second coefficient of viscosity. For an incompressible fluid we have 1.3, and consequently the above relation reduces to *Stoke's Law* (4):

$$\sigma = -p\mathbf{I} + \mu(\nabla\mathbf{v} + \nabla\mathbf{v}^T) \quad (1.7)$$

The second term of 1.4 can be written as:

$$\nabla.(\rho\mathbf{v}\mathbf{v}) = \rho\mathbf{v}.\nabla\mathbf{v} + \rho(\nabla.\mathbf{v})\mathbf{v}$$

that for an incompressible fluid (equation 1.3) results in:

$$\nabla.(\rho\mathbf{v}\mathbf{v}) = \rho\mathbf{v}.\nabla\mathbf{v}$$

Using this last result and the *Stoke's Law*, the equation of motion can be written as:

$$\rho\frac{\partial\mathbf{v}}{\partial t} + \rho\mathbf{v}.\nabla\mathbf{v} = -\nabla p + \mu\nabla^2\mathbf{v} + \rho\mathbf{g}$$

and therefore we have:

$$\frac{\partial\mathbf{v}}{\partial t} + \mathbf{v}.\nabla\mathbf{v} = \frac{-\nabla p}{\rho} + \nu\nabla^2\mathbf{v} + \mathbf{g} \quad (1.8)$$

where  $\nu = \frac{\mu}{\rho}$  is the fluid kinematic viscosity. This equation is called the Navier-Stokes equation.

### 1.2.1 Adimensionalization

The equation 1.8 is written in dimensional form. For this, the following scales for the variables are used (6):

$$\mathbf{v} = U\mathbf{v}^* \quad x = Lx^*$$

$$p = \rho U^2 p^* \quad t = \frac{L}{U} t^*$$

where the star indicates adimensional variables. Substituting into the Navier-Stokes equations (1.8) we have:

$$\frac{\partial \mathbf{v}^*}{\partial t^*} + \mathbf{v}^* \cdot \nabla \mathbf{v}^* = -\nabla p^* + \frac{1}{\frac{UL}{\nu}} \nabla^2 \mathbf{v}^* + \frac{gL}{U^2} \frac{\mathbf{g}}{g} \quad (1.9)$$

The adimensional group  $Re = \frac{UL}{\nu}$  is the fluid Reynolds number. It depends of the fluid physical properties and the geometrical properties of the domain. The adimensional group  $Fr = \frac{U}{\sqrt{Lg}}$  is called the fluid Froude number. Using this adimensional numbers and considering every variable as adimensional, we obtain the adimensional form of the Navier-Stokes equation (6):

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{v} + \frac{1}{Fr^2} \frac{\mathbf{g}}{g} \quad (1.10)$$

### 1.3 The convection-diffusion equation

Let  $\phi$  be a scalar quantity of a property of the fluid (like temperature or concentration of a substance, like salt). The transport equation of scalar quantity  $\phi$  is governed by (6):

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi + \nabla \cdot \sigma = f \quad (1.11)$$

where  $\mathbf{v}$  is the fluid velocity and  $f$  is the forcing term. In this equation we have already have considered incompressibility ( $\nabla \cdot \mathbf{v} = 0$ ). Considering that the diffusive flux is given by *Fick's first law*, i.e.:

$$\sigma = -\alpha \nabla \phi \quad (1.12)$$

(where  $\alpha$  is a diffusivity constant) then (6):

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi - \alpha \nabla^2 \phi = f \quad (1.13)$$

This equation describes transport of a substance in a fluid.

### 1.3.1 Adimensionalization

The convection-diffusion equation adimensionalization leads to a adimensional quantity: the *Péclet* number. As we have done in the previous section, if we take (6):

$$\mathbf{v} = U \mathbf{v}^* \quad x = x^* L$$

$$\phi = \phi_0 \phi^* \quad t = \frac{L}{U} t^*$$

and substitute in equation 1.13:

$$\frac{\partial \phi^*}{\partial t^*} + \mathbf{v}^* \cdot \nabla \phi^* - \frac{\alpha}{UL} \nabla^2 \phi^* = f \quad (1.14)$$

The adimensional group  $Pe = \frac{UL}{\alpha}$  its called the *Péclet number*. Using this definition and taking every variable as adimensional, the equation before is left in adimensional form:

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi - \frac{1}{Pe} \nabla^2 \phi = f \quad (1.15)$$

## 1.4 Boundary and initial conditions

Once the governing equations are defined, boundary and initial conditions (the latter only in transient problems) must be adequately prescribed in order to close the problem. Boundaries can be classified, according to the physics of the boundary, as (8):

- Fictitious boundaries
- Real boundaries

A typical open domain describing flow past an aircraft wing is an example of an fictitious boundary, taken as limits of computation. With suitable values specified at such boundaries, however, accurate solution for the flow inside the isolated domain can be achieved. If the distance to the object (e.g. the aircraft wing) is far enough, the boundary values tend to those encountered in the free domain flow (the so-called flow at infinity). In subsonic flows (as in this work) all quantities except the density can be made at such boundaries (8).

Real boundaries are those of real physical limits of the fluid domain and there are three different possibilities (8):

1. *Solid boundaries with no slip conditions*: the fluid is assumed to stick itself to the boundary and thus velocity at that point becomes zero. Only possible for viscous fluids.
2. *Solid boundaries in inviscid flow (slip condition)*: When the flow is inviscid we will always encounter slipping boundary conditions. Only the normal velocity component is specified, normally zero (if the boundary is motionless)
3. *Prescribed traction boundary conditions*: tractions are prescribed. This includes zero traction in the case of free surface fluids or any prescribed tractions such as those caused by wind being imposed on the surface.

Another classification of the boundary conditions can be (4):

1. *Dirichlet boundary conditions*: prescribe the value of the unknown function along the boundary.
2. *Neumann boundary conditions*: imposes the normal gradient of the unknown function along the boundary.
3. *Robin boundary conditions*: are a combination of the Dirichlet and Neumann conditions.

In this work we will only use boundary conditions of the first two types above. If  $\Omega$  is the problem domain in  $[0, T]$ , being  $T$  the end of the time interval used in the computations, and  $\Gamma = \Gamma_g \cup \Gamma_h$  with  $\Gamma_g \cap \Gamma_h = \emptyset$  (a partition of  $\Gamma$ , being  $\Gamma$  the domain boundary), then essential (Dirichlet) and natural (Neumann) boundary conditions for the Navier-Stokes equations are:

$$\mathbf{v} = v_g \text{ on } \Gamma_g \times [0, T] \quad (1.16)$$

$$\nabla \cdot \mathbf{v} \mathbf{n} = h \text{ on } \Gamma_h \times [0, T] \quad (1.17)$$

where  $n$  is the unit outward normal vector of  $\Gamma$ . The initial condition for the incompressible Navier-Stokes equations is a divergence free velocity field  $\mathbf{v}_0(\mathbf{x})$  specified over the domain:

$$\mathbf{v}(\mathbf{x}, 0) = \mathbf{v}_0(\mathbf{x}) \quad (1.18)$$

Similarly, considering another partition of the boundary  $\Gamma = \Gamma_r \oplus \Gamma_s$ , boundary conditions for the convection-diffusion equations can be stated as:

$$\phi = r \text{ on } \Gamma_r \times [0, T] \quad (1.19)$$

$$\nabla \phi \cdot \mathbf{n} = s \text{ on } \Gamma_s \times [0, T] \quad (1.20)$$

being  $r$  and  $s$  two given scalar functions. As initial condition, an scalar field  $\phi_0(x)$  is specified over the domain  $\Omega_t$  at  $t = 0$ :

$$\phi(x, 0) = \phi_0(x) \quad (1.21)$$

## 2 DISCRETIZATION METHOD AND SUPG FORMULATION

In this chapter we will present the finite element formulation, beginning with the description of the variational form of the Navier-Stokes equations, the semi-discrete Galerkin formulation, i.e., the spatial discretization of the partial differential equations, and the Streamline Upwind/Petrov Galerkin (SUPG) method, which is an stabilized method based in a new term included in the equation, involving the equation residual. Afterwards, the solution method for the resulting system of equations will be shown.

### 2.1 Variational formulation

The fluid flow solution is given by the equations  $\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$ ,  $p = p(\mathbf{x}, t)$  and  $\phi(\mathbf{x}, t)$  defined in  $\Omega \times [0, T]$  (being  $\Omega$  the problem domain and  $T$  the total time), which satisfy the following differential equation system:

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{v} + \frac{1}{Fr^2} \frac{\mathbf{g}}{g} \quad (2.1)$$

$$\nabla \cdot \mathbf{v} = 0 \quad (2.2)$$

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi - \frac{1}{Pe} \nabla^2 \phi = f \quad (2.3)$$

restricted to the following boundary conditions:

$$\mathbf{v} = v_g \text{ on } \Gamma_g \times [0, T] \quad (2.4)$$

$$\nabla \mathbf{v} \cdot \mathbf{n} = h \text{ on } \Gamma_h \times [0, T] \quad (2.5)$$

$$\phi = r \text{ on } \Gamma_r \times [0, T] \quad (2.6)$$

$$\nabla \phi \cdot \mathbf{n} = s \text{ on } \Gamma_s \times [0, T] \quad (2.7)$$

where  $v_g$ ,  $v_h$ ,  $r$  and  $s$  are given functions,  $\Gamma_g$ ,  $\Gamma_h$  and  $\Gamma_r$ ,  $\Gamma_s$  are two partitions of the domain boundary  $\Gamma$ . Initial conditions are:

$$\mathbf{v}(\mathbf{x}, 0) = \mathbf{v}_0(\mathbf{x}) \quad (2.8)$$

$$\phi(x, 0) = \phi_0(x) \quad (2.9)$$

Considering the subspace:

$$V = H^1(\Omega)^m = \{\mathbf{v} = (v_1, \dots, v_m) : v_i \in H^1(\Omega), \forall i = 1, \dots, m\} \quad (2.10)$$

( $m$  is the spatial dimension) where  $H^1(\Omega)$  is the Sobolev space given by:

$$H^1(\Omega) = \left\{ v \in L^2(\Omega) : \frac{\partial v}{\partial x_i} \in L^2, i = 1, \dots, m \right\} \quad (2.11)$$

we will define:

$$V_\Gamma = \{\mathbf{v} \in V : \mathbf{v} = v_g \text{ on } \Gamma_g\} \quad (2.12)$$

$$V_0 = \{\mathbf{v} \in V : \mathbf{v} = 0 \text{ on } \Gamma_g\} \quad (2.13)$$

$$Y_\Gamma = \{\phi \in H^1(\Omega) : \phi = r \text{ on } \Gamma_r\} \quad (2.14)$$

$$Y_0 = \{\phi \in H^1(\Omega) : \phi = 0 \text{ on } \Gamma_r\} \quad (2.15)$$

$$P = \{q \in L^2(\Omega)\} \quad (2.16)$$

The variational formulation (Galerkin) consist in finding solutions  $\mathbf{v}(\mathbf{x}, t) \in V_\Gamma$ ,  $p \in P$  such that:

$$\int_{\Omega} \left\{ \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p - \frac{1}{Re} \nabla^2 \mathbf{v} - \frac{1}{Fr^2 g} \mathbf{g} \right\} w d\Omega = 0 \quad (2.17)$$

$$\int_{\Omega} [\nabla \cdot \mathbf{v}] q d\Omega = 0 \quad (2.18)$$

where  $w$  are the test functions,  $w \in V_0$ . And for the convection-diffusion equation:

$$\int_{\Omega} \left\{ \frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi - \frac{1}{Pe} \nabla^2 \phi - f \right\} w d\Omega = 0 \quad (2.19)$$

where  $w \in Y_0$ . Applying integration by parts we have:

$$\int_{\Omega} \nabla^2 \mathbf{v} w d\Omega = - \int_{\Omega} \nabla \mathbf{v} \nabla w d\Omega + \int_{\Gamma} w \nabla \mathbf{v} n d\Gamma \quad (2.20)$$

$$\int_{\Omega} \nabla^2 \phi d\Omega = - \int_{\Omega} \nabla \phi \nabla w d\Omega + \int_{\Gamma} w \nabla \phi n d\Gamma \quad (2.21)$$

Substituting these expressions in 2.17 and 2.19 and considering two facts: first, that the test functions  $w$  are zero on the  $\Gamma_g$  ( $\Gamma_r$ ) part of the boundary; and second, the boundary conditions 2.5 and 2.7:

$$\int_{\Gamma} w \nabla \mathbf{v} n d\Gamma = \int_{\Gamma_h} w \nabla \mathbf{v} n d\Gamma = \int_{\Gamma_h} w h d\Gamma \quad (2.22)$$

$$\int_{\Gamma} w \nabla \phi n d\Gamma = \int_{\Gamma_s} w \nabla \phi n d\Gamma = \int_{\Gamma_s} w s d\Gamma \quad (2.23)$$

Thus, we can obtain the following expressions:

$$\begin{aligned} \int_{\Omega} \left\{ \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p \right\} w d\Omega + \int_{\Omega} \frac{1}{Re} \nabla \mathbf{v} \nabla w d\Omega = \\ \int_{\Omega} \frac{1}{Fr^2} \frac{\mathbf{g}}{g} w d\Omega + \int_{\Gamma_h} \frac{1}{Re} w h d\Gamma \end{aligned} \quad (2.24)$$

$$\int_{\Omega} \left\{ \frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi \right\} w d\Omega + \int_{\Omega} \frac{1}{Pe} \nabla \phi \cdot \nabla w d\Omega = \int_{\Omega} f w d\Omega + \int_{\Gamma_s} w s d\Gamma \quad (2.25)$$

## 2.2 Streamline Upwind/ Petrov Galerkin method

Solutions computed with the classical Galerkin method are often corrupted by spurious node-to-node oscillations. These oscillations are more likely to appear in convection dominated cases (high Peclet or Reynolds number) (2). SUPG-type methods are Galerkin schemes with additional correction terms to better capture the convective effects (9). Originally, *upwind* methods were developed to overcome this problem, but these upwind methods exhibit excessive diffusion perpendicular to the flow direction, corrupting the true solution. The SUPG method was designed to get over this, it only adds diffusion in the flow direction. This method was first introduced by Hughes and Brooks (2), who studied the convection-diffusion equation as a model problem for the incompressible Navier-Stokes equations at high Reynolds numbers (9). If we are solv-

ing the classical Galerkin approach, in equations 2.24 and 2.25  $w \in V_0$ , but in the SUPG method, test and trial functions do not belong to the same spaces. SUPG test functions are defined as  $w_{SUPG} = w + \tau_{SUPG} \mathbf{v} \cdot \nabla w$ .  $\tau_{SUPG}$  is an stabilization parameter that quantifies the additional diffusion added.  $\tau_{SUPG}$  definition will be discussed in the next section.

### 2.2.1 The $\tau_{SUPG}$ stabilization parameter

Several definitions of  $\tau_{SUPG}$  parameter can be found in literature. The first definition, found in (2), is:

$$\tau_{SUPG} = \sum \frac{\varepsilon_i v_i h_i}{\sqrt{15}} \quad (2.26)$$

$$\varepsilon_i = \coth \alpha_i - \frac{1}{\alpha_i} \quad (2.27)$$

$$\alpha_i = \frac{\rho v_i h_i}{2\mu} \quad (2.28)$$

$\rho$  and  $\mu$  are the fluid's density and kinematic viscosity respectively;  $v_i$ ,  $h_i$  the velocity and the element's size in the direction  $i$ . This  $\tau_{SUPG}$  definition was restricted to quadrilateral elements, and can easily be extended to hexaedral elements. Following Tezduyar (10), we make use of the UGN-based stabilization parameters:

$$\tau_1 = \left( \sum_{a=1}^{n_{en}} |\mathbf{v}_a \cdot \nabla N_a| \right)^{-1} \quad (2.29)$$

$$\tau_2 = \frac{\Delta t}{2} \quad (2.30)$$

$$\tau_3 = \frac{h_{UGN}^2}{4\nu} \quad (2.31)$$

where  $n_{en}$  is the number of element nodes,  $\nu$  is the fluid kinematic viscosity and  $N_a$  is the interpolation function associated with node  $a$ , and the *element length*  $h_{UGN}$  is defined as:

$$h_{UGN} = 2\|\mathbf{v}^h\|\tau_1 \quad (2.32)$$

We can now define  $\tau_{SUPG}$  as follows (10):

$$\tau_{SUPG} = \left( \frac{1}{\tau_1^2} + \frac{1}{\tau_2^2} + \frac{1}{\tau_3^2} \right)^{-\frac{1}{2}} \quad (2.33)$$

These new definitions automatically take into account the local length scales, advection field and the element-level Reynolds and CFL (Courant-Friedrich-Lewis) numbers (10). CFL number is defined as:

$$CFL = \frac{U \Delta t}{h_{elem}} \quad (2.34)$$

where  $U$  is the fluid velocity,  $\Delta t$  is the time step and  $h_{elem}$  is the element characteristic size, that has many definitions: in this work it will be taken as the diameter of the circumcircle of the element (we have used two types of meshes: structured with quadrilateral elements, and unstructured with triangular elements, so the elements will always have an circumcircle).

## 2.3 Discretization and solution method

### 2.3.1 Spatial discretization

The standard Galerkin approximation consists in decomposing the domain  $\Omega$  into  $N_{el}$  elements such that they cover the domain. The spaces  $V$ ,  $V_0$  (and  $Y$ ,  $Y_0$  for the convection-diffusion equation) and  $P$  are approached by finite dimensional spaces  $V^h$ ,  $V_0^h$  ( $Y^h$ ,  $Y_0^h$ ) and  $P^h$  spanned by continuous piecewise polynomials, where functions  $\mathbf{v}_h \in V^h$ ,  $p_h \in P^h$  and  $\phi_h \in Y^h$  are:

$$\mathbf{v}_h(\mathbf{x}, t) = \sum_{n=1}^{NV} w_n(\mathbf{x}) \mathbf{v}_n(t) \quad (2.35)$$

$$p_h(\mathbf{x}, t) = \sum_{n=1}^{NP} q_n(\mathbf{x}) p_n(t) \quad (2.36)$$

$$\phi_h(\mathbf{x}, t) = \sum_{n=1}^{NC} c_n(\mathbf{x}) \phi_n(t) \quad (2.37)$$

where  $NV$ ,  $NP$  and  $NC$  are the number of velocity, pressure and concentration points in the domain discretization. If we take  $\mathbf{v}(\mathbf{x}, t) = \mathbf{v}_h(\mathbf{x}, t)$ ,  $p(\mathbf{x}, t) = p_h(\mathbf{x}, t)$  and  $\phi(\mathbf{x}, t) = \phi_h(\mathbf{x}, t)$  into equations 2.18, 2.24 and 2.25 the semi-discrete form (because it is still continuous in time) is obtained:

$$\int_{\Omega} \left\{ \frac{\partial \mathbf{v}_h}{\partial t} + \mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h \right\} w'_h d\Omega + \int_{\Omega} \frac{1}{Re} \nabla \mathbf{v}_h \nabla w'_h d\Omega = \\ \int_{\Omega} \frac{1}{Fr^2} \frac{\mathbf{g}}{g} w'_h d\Omega + \int_{\Gamma_h} \frac{1}{Re} w' h d\Gamma \quad (2.38)$$

$$\int_{\Omega} [\nabla \cdot \mathbf{v}_h] q_h d\Omega = 0 \quad (2.39)$$

$$\int_{\Omega} \left\{ \frac{\partial \phi_h}{\partial t} + \mathbf{v} \cdot \nabla \phi_h \right\} c'_h d\Omega + \int_{\Omega} \frac{1}{Pe} \nabla \phi_h \cdot \nabla c'_h d\Omega = \int_{\Omega} f c'_h d\Omega + \int_{\Gamma_h} c'_h s d\Gamma \quad (2.40)$$

with  $w' = w_{SUPG} = w + \tau_{SUPG} \mathbf{v} \cdot \nabla w$ , and the same definition for  $c'$ .

### 2.3.2 Newton-Krylov solution method

As can be seen in section 2.3.1 , a nonlinear system of equations is generated, due to the convective term in the Navier-Stokes equations. In this work, we will make use of a Newton-Krylov method. The underlying principle of this method is to minimize the residual in a Krylov space by linearizing the equation using the Newton method and solve the resulting linear system of algebraic equations with a Krylov method. Since the Jacobian matrix is neither formed nor stored, the Newton-Krylov method can be specially referred to as a matrix-free Newton-Krylov method (11).

Considering a function  $st$  (product of the functions  $s$  and  $t$ ), we can expand it in a Taylor series about the current value and terminate the series expansion after the first derivative terms. The results is as follows:

$$s^{k+1}t^{k+1} = s^k t^k + \left[ \frac{\partial}{\partial s} (st)^k \right] (s^{k+1} - s^k) + \left[ \frac{\partial}{\partial t} (st)^k \right] (t^{k+1} - t^k) + H.O.T. \quad (2.41)$$

$$= s^{k+1}t^k + s^k t^{k+1} - s^k t^k + H.O.T. \quad (2.42)$$

with  $k$  being a certain step in an iterative process to obtain a solution, and *H.O.T* stands for *higher order terms*. So, applying 2.41 to the convective term of the Navier-Stokes equation we reach to:

$$(\mathbf{v}^{k+1} \nabla) \mathbf{v}^{k+1} = (\mathbf{v}^{k+1} \cdot \nabla) \mathbf{v}^k + (\mathbf{v}^k \cdot \nabla) \mathbf{v}^{k+1} - (\mathbf{v}^k \cdot \nabla) \mathbf{v}^k \quad (2.43)$$

$k$  values are taken from the previous iteration, and  $k + 1$  values are unknowns.

Going back to equations 2.38, 2.39 and 2.40, we can state the following:

$$w'_h = \sum_{n=1}^{NV} \alpha_n w'_n \quad (2.44)$$

$$q_h = \sum_{n=1}^{NP} \alpha_n q_n \quad (2.45)$$

$$c'_h = \sum_{n=1}^{NC} \alpha_n c'_n \quad (2.46)$$

Replacing 2.44 in 2.38:

$$\begin{aligned} & \int_{\Omega} \left( \frac{\partial \mathbf{v}_h}{\partial t} + \mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h \right) \left( \sum_{n=1}^{NV} \alpha_n w'_n \right) d\Omega + \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v}_h \nabla \left( \sum_{n=1}^{NV} \alpha_n w'_n \right) d\Omega \\ & - \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} \left( \sum_{n=1}^{NV} \alpha_n w'_n \right) d\Omega - \frac{1}{Re} \int_{\Gamma_h} h \left( \sum_{n=1}^{NV} \alpha_n w'_n \right) d\Gamma = 0 \end{aligned}$$

Rearranging terms:

$$\begin{aligned} & \alpha_1 \left\{ \int_{\Omega} \left( \frac{\partial \mathbf{v}_h}{\partial t} + \mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h \right) w'_1 d\Omega + \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v}_h \nabla w'_1 d\Omega - \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w'_1 d\Omega \right. \\ & \quad \left. - \frac{1}{Re} \int_{\Gamma_h} h w'_1 d\Gamma \right\} \\ & + \alpha_2 \left\{ \int_{\Omega} \left( \frac{\partial \mathbf{v}_h}{\partial t} + \mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h \right) w'_2 d\Omega + \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v}_h \nabla w'_2 d\Omega - \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w'_2 d\Omega \right. \\ & \quad \left. - \frac{1}{Re} \int_{\Gamma_h} h w'_2 d\Gamma \right\} + \dots \\ & + \alpha_{NV} \left\{ \int_{\Omega} \left( \frac{\partial \mathbf{v}_h}{\partial t} + \mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h \right) w'_{NV} d\Omega + \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v}_h \nabla w'_{NV} d\Omega - \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w'_{NV} d\Omega \right. \\ & \quad \left. - \frac{1}{Re} \int_{\Gamma_h} h w'_{NV} d\Gamma \right\} = 0 \end{aligned}$$

This last expression must be valid for every  $w'_h$ , hence the terms that multiply  $\alpha_1, \alpha_2 \dots \alpha_{NV}$  must be zero:

$$\begin{aligned}
& \int_{\Omega} \left( \frac{\partial \mathbf{v}_h}{\partial t} + \mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h \right) w'_1 d\Omega + \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v}_h \nabla w'_1 d\Omega - \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w'_1 d\Omega \\
& \quad - \frac{1}{Re} \int_{\Gamma_h} h w'_1 d\Gamma = 0 \\
& \int_{\Omega} \left( \frac{\partial \mathbf{v}_n}{\partial t} + \mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h \right) w'_2 d\Omega + \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v}_h \nabla w'_2 d\Omega - \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w'_2 d\Omega \\
& \quad - \frac{1}{Re} \int_{\Gamma_h} h w'_2 d\Gamma = 0 \\
& \quad \vdots \\
& \int_{\Omega} \left( \frac{\partial \mathbf{v}_h}{\partial t} + \mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h \right) w'_{NV} d\Omega + \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v}_h \nabla w'_{NV} d\Omega - \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w'_{NV} d\Omega \\
& \quad - \frac{1}{Re} \int_{\Gamma_h} h w'_{NV} d\Gamma = 0
\end{aligned}$$

Making use of 2.35 and 2.36 and the linearization 2.43 in these last equations we obtain:

$$\begin{aligned}
& \int_{\Omega} \left( \frac{\partial (\sum_{n=1}^{NV} w_n(\mathbf{x}) \mathbf{v}_n(t))}{\partial t} + ((\sum_{n=1}^{NV} w_n(\mathbf{x}) \mathbf{v}_n(t)).\nabla) \mathbf{v}^k + (\mathbf{v}^k.\nabla)(\sum_{n=1}^{NV} w_n(\mathbf{x}) \mathbf{v}_n(t)) - \right. \\
& \quad \left. (\mathbf{v}^k.\nabla) \mathbf{v}^k + \nabla(\sum_{n=1}^{NP} q_n(\mathbf{x}) p_n(t)) \right) w'_j d\Omega + \frac{1}{Re} \int_{\Omega} \nabla(\sum_{n=1}^{NV} w_n(\mathbf{x}) \mathbf{v}_n(t)) \nabla w'_j d\Omega \\
& = \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w'_j d\Omega + \frac{1}{Re} \int_{\Gamma_h} h w'_j d\Gamma
\end{aligned}$$

where  $j = 1, \dots, NV$  and  $\mathbf{v}^k$  is the fluid velocity computed in the previous iteration (or the initial conditions, in the case of the first iteration). Rearranging some terms we obtain:

$$\begin{aligned}
& \int_{\Omega} \left( \sum_{n=1}^{NV} w_n \frac{\partial \mathbf{v}_n}{\partial t} + ((\sum_{n=1}^{NV} w_n \mathbf{v}_n).\nabla) \mathbf{v}^k + \sum_{n=1}^{NV} w_n (\mathbf{v}^k.\nabla) (\mathbf{v}_n) - (\mathbf{v}^k.\nabla) \mathbf{v}^k \right. \\
& \quad \left. + \sum_{n=1}^{NP} p_n \nabla q_n \right) w'_j d\Omega + \sum_{n=1}^{NV} \frac{1}{Re} \int_{\Omega} \mathbf{v}_n \nabla w_n \nabla w'_j d\Omega \\
& = \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w'_j d\Omega + \frac{1}{Re} \int_{\Gamma_h} h w'_j d\Gamma
\end{aligned}$$

Similarly, using 2.36 and 2.45 in 2.38 we have :

$$\sum_{n=1}^{NV} \int_{\Omega} (\nabla \cdot (w_n \mathbf{v}_n)) q_j d\Omega = 0$$

where  $j = 1, \dots, NP$ . And now, making use of 2.46 and 2.37 in 2.40, the following result is attained:

$$\sum_{n=1}^{NC} \int_{\Omega} \left\{ w_n \frac{\partial \phi_n}{\partial t} + (\mathbf{v} \cdot \nabla w_n) \phi_n \right\} w'_j d\Omega + \sum_{n=1}^{NC} \int_{\Omega} \phi_n \frac{1}{Pe} \nabla w_n \cdot \nabla w'_j d\Omega = \int_{\Omega} f w'_j d\Omega + \int_{\Gamma_h} w'_j s d\Gamma$$

with  $j = 1, \dots, NV$ . Taking into consideration that the domain  $\Omega$  is divided into  $N_{el}$  elements, the three equations above can be reformulated as:

Navier-Stokes:

$$\begin{aligned} & \sum_{l=1}^{N_{el}} \int_{\Omega_e} \left( \sum_{n=1}^{NV} w_n \frac{\partial \mathbf{v}_n}{\partial t} + ((\sum_{n=1}^{NV} w_n \mathbf{v}_n) \cdot \nabla) \mathbf{v}^k + \sum_{n=1}^{NV} w_n (\mathbf{v}^k \cdot \nabla) (\mathbf{v}_n) - (\mathbf{v}^k \cdot \nabla) \mathbf{v}^k \right. \\ & \quad \left. + \sum_{n=1}^{NP} p_n \nabla q_n \right) w'_j d\Omega + \sum_{l=1}^{N_{el}} \sum_{n=1}^{NV} \frac{1}{Re} \int_{\Omega_e} \mathbf{v}_n \nabla w_n \nabla w'_j d\Omega \\ & \quad = \frac{1}{Fr^2} \sum_{l=1}^{N_{el}} \int_{\Omega_e} \frac{\mathbf{g}}{g} w'_j d\Omega + \sum_{l=1}^{N_{el}} \frac{1}{Re} \int_{\Gamma_{h_e}} h w'_j d\Gamma \end{aligned}$$

Continuity:

$$\sum_{l=1}^{N_{el}} \sum_{n=1}^{NV} \int_{\Omega_e} (\nabla \cdot (w_n \mathbf{v}_n)) q_j d\Omega = 0$$

Scalar transport:

$$\begin{aligned} & \sum_{l=1}^{N_{el}} \sum_{n=1}^{NC} \int_{\Omega_e} \left\{ w_n \frac{\partial \phi_n}{\partial t} + (\mathbf{v} \cdot \nabla w_n) \phi_n \right\} w'_j d\Omega + \sum_{l=1}^{N_{el}} \sum_{n=1}^{NC} \int_{\Omega_e} \phi_n \frac{1}{Pe} \nabla w_n \cdot \nabla w'_j d\Omega \\ & \quad = \sum_{l=1}^{N_{el}} \int_{\Omega_e} f w'_j d\Omega + \sum_{l=1}^{N_{el}} \int_{\Gamma_{h_e}} w'_j s d\Gamma \end{aligned}$$

In these three last equations, the unknowns are the  $\mathbf{v}_i$ , the  $p_i$  and the  $\phi_i$ , which leads to the *matrix form* of the equations, seen in the next section.

### 2.3.3 Matrix form of the equations

The spatial discretization of the domain, leads to a matrix form of the Navier-Stokes and convection-diffusion equations. The solution vector is  $\mathbf{v} = (v_{11}, \dots, v_{1NV}, v_{21}, \dots, v_{2NV}, \dots, v_{m1}, \dots, v_{mNV})$  if the number of spatial dimensions is  $m$ , where  $\mathbf{v}_j = (v_{1j}, v_{2j}, \dots, v_{mj})$  is the fluid velocity at any given point  $j$ .  $\mathbf{p} = (p_1, \dots, p_{NP})$  and  $\phi = (\phi_1, \dots, \phi_{NC})$ . Considering this, we have:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{K}\mathbf{v} - \mathbf{G}\mathbf{p} = \mathbf{F} \quad (2.47)$$

$$\mathbf{G}'\mathbf{v} = \mathbf{D} \quad (2.48)$$

$$\mathbf{M}_c\dot{\phi} + \mathbf{C}_c\phi = \mathbf{F}_c \quad (2.49)$$

$\mathbf{D}$  is a generalized force vector representing the effect of the prescribed boundary velocities. Matrix  $\mathbf{K}$  can be divided into the sum of three matrices:  $K_1$ ,  $K_2$  and  $K_3$ , the first one being the contribution of the diffusive term, and the other two, from the convective linearized terms. The matrices above can be block-divided for better understanding:

$$\mathbf{M}_\rho = \begin{bmatrix} M_x & 0 & 0 \\ 0 & M_y & 0 \\ 0 & 0 & M_z \end{bmatrix}_{3NV \times 3NV} \quad (2.50a)$$

$$\mathbf{K}_1 = \begin{bmatrix} K_{1_x} & 0 & 0 \\ 0 & K_{1_y} & 0 \\ 0 & 0 & K_{1_z} \end{bmatrix}_{3NV \times 3NV} \quad (2.50b)$$

$$\mathbf{K}_2 = \begin{bmatrix} K_{2_x} & 0 & 0 \\ 0 & K_{2_y} & 0 \\ 0 & 0 & K_{2_z} \end{bmatrix}_{3NV \times 3NV} \quad (2.50c)$$

$$\mathbf{K}_3 = \begin{bmatrix} K_{3_{xx}} & K_{3_{xy}} & K_{3_{xz}} \\ K_{3_{yx}} & K_{3_{yy}} & K_{3_{yz}} \\ K_{3_{zx}} & K_{3_{zy}} & K_{3_{zz}} \end{bmatrix}_{3NV \times 3NV} \quad (2.50d)$$

$$\mathbf{F} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}_{NP \times 3NV} \quad (2.50e)$$

$$\mathbf{D} = \begin{bmatrix} D_x & D_y & D_z \end{bmatrix}_{NP \times 3NV} \quad (2.50f)$$

$$\mathbf{G} = \begin{bmatrix} G_x \\ G_y \\ G_z \end{bmatrix}_{3NV \times NP} \quad (2.50g)$$

$$\mathbf{G}' = \begin{bmatrix} G_x & G_y & G_z \end{bmatrix}_{NP \times 3NV} \quad (2.50h)$$

$$\mathbf{M}_c = \begin{bmatrix} M_c \end{bmatrix}_{NC\theta \times NC\theta} \quad (2.50i)$$

$$\mathbf{C}_c = \begin{bmatrix} C_c \end{bmatrix}_{NC\theta \times NC\theta} \quad (2.50j)$$

$$\mathbf{F}_c = \begin{bmatrix} C_c \end{bmatrix}_{NC\theta \times 1} \quad (2.50k)$$

To avoid introducing more nonlinearities in the model, the fluid velocity  $\mathbf{v}$  in the SUPG term  $\tau_{SUPG}\mathbf{v} \cdot \nabla w$  is taken as latest value computed, and thus, it will be a known variable for us. Taking this into account, the matrices terms are:

$$[M_{x_{ij}}] = [M_{y_{ij}}] = [M_{z_{ij}}] = \int_{\Omega_e} (w_i + \tau_{SUPG} \mathbf{v} \cdot \nabla w_i) w_j d\Omega \quad (2.51a)$$

$$[K_{1x_{ij}}] = [K_{1y_{ij}}] = [K_{1z_{ij}}] = -\frac{1}{Re} \int_{\Omega_e} (w_i + \tau_{SUPG} \mathbf{v} \cdot \nabla w_i) \nabla^2 w_j d\Omega \quad (2.51b)$$

$$[K_{2x_{ij}}] = [K_{2y_{ij}}] = [K_{2z_{ij}}] = \int_{\Omega_e} (w_i + \tau_{SUPG} \mathbf{v} \cdot \nabla w_i) \mathbf{v} \cdot \nabla w_j d\Omega \quad (2.51c)$$

$$[K_{3kl_{ij}}] = \int_{\Omega_e} (w_i + \tau_{SUPG} \mathbf{v} \cdot \nabla w_i) \left( \frac{\partial v_k}{\partial x_l} \right) w_j d\Omega \quad (2.51d)$$

$$[F_{x_i}] = [F_{y_i}] = [F_{z_i}] = \int_{\Omega_e} (w_i + \tau_{SUPG} \mathbf{v} \cdot \nabla w_i) \left( \frac{1}{Fr^2} \frac{\mathbf{g}}{g} \right) d\Omega \quad (2.51e)$$

$$[G_{x_{ij}}] = [G_{y_{ij}}] = [G_{z_{ij}}] = \int_{\Omega_e} (w_i + \tau_{SUPG} \mathbf{v} \cdot \nabla w_i) \nabla q_j d\Omega \quad (2.51f)$$

$$[G'_{x_{ij}}] = [G'_{y_{ij}}] = [G'_{z_{ij}}] = \int_{\Omega_e} q_i \nabla w_j d\Omega \quad (2.51g)$$

$$[M_{c_{ij}}] = \int_{\Omega_e} (w_i + \tau_{SUPG} \mathbf{v} \cdot \nabla w_i) w_j d\Omega \quad (2.51h)$$

$$[C_{c_{ij}}] = \int_{\Omega_e} (w_i + \tau_{SUPG} \mathbf{v} \cdot \nabla w_i) \left( \mathbf{v} \cdot \nabla w_j - \frac{1}{Pe} \nabla^2 w_j \right) d\Omega \quad (2.51i)$$

$$[F_{c_i}] = \int_{\Omega_e} f (w_i + \tau_{SUPG} \mathbf{v} \cdot \nabla w_i) d\Omega \quad (2.51j)$$

### 2.3.4 Discretization in time

To solve the equations 2.47, 2.48 and 2.49 we make use of an *theta scheme*, in order to generalize results: from that point on we can implement a backward Euler, forward Euler or Crank-Nicolson scheme very simply.

Lets consider the following time discretization:

$$\dot{\mathbf{v}} = \frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\Delta t} \quad (2.52a)$$

$$\dot{\phi} = \frac{\phi^{n+1} - \phi^n}{\Delta t} \quad (2.52b)$$

where  $n$  indicates a certain time step, and the *theta scheme* equations:

$$\mathbf{v} = \theta \mathbf{v}^{n+1} + (1 - \theta) \mathbf{v}^n \quad (2.53a)$$

$$p = p^{n+1} + (1 - \theta) p^n \quad (2.53b)$$

$$\phi = \phi^{n+1} + (1 - \theta) \phi^n \quad (2.53c)$$

Hence, if these equations before are taken into 2.47, 2.48 and 2.49, we get the following result:

$$(\mathbf{M} + \Delta t \theta \mathbf{K}) \mathbf{v}_{k+1}^{n+1} - \Delta t \theta \mathbf{G} p_{k+1}^{n+1} = (\mathbf{M} + \Delta t \theta \mathbf{K}) \mathbf{v}_k^n + \Delta t(1 - \theta) \mathbf{G} p_k^n + \mathbf{F} \quad (2.54a)$$

$$\theta \mathbf{G}' \mathbf{v}_{k+1}^{n+1} = -(1 - \theta) \mathbf{G}' \mathbf{v}_k^n + \mathbf{D} \quad (2.54b)$$

$$(\mathbf{M}_c + \Delta t \theta \mathbf{C}_c) \phi^{n+1} = (\mathbf{M}_c - \Delta t(1 - \theta) \mathbf{C}_c) \phi^n + \mathbf{F}_c \quad (2.54c)$$

this last equation system can be written as:

$$\begin{bmatrix} \mathbf{M} + \Delta t \theta \mathbf{K} & -\Delta t \theta \mathbf{G} \\ -\Delta t \theta \mathbf{G}' & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_{k+1}^{n+1} \\ \mathbf{p}_{k+1}^{n+1} \end{bmatrix} = \begin{bmatrix} (\mathbf{M} - \Delta t(1 - \theta) \mathbf{K}) \mathbf{v}_k^n + \Delta t(1 - \theta) \mathbf{G} \mathbf{p}_k^n + \mathbf{F} \\ \Delta t(1 - \theta) \mathbf{G} \mathbf{v}_k^n - \Delta t D \end{bmatrix} \quad (2.55a)$$

$$\begin{bmatrix} \mathbf{M}_c + \Delta t \theta \mathbf{C}_c \\ \mathbf{M}_c - \Delta t(1 - \theta) \mathbf{C}_c \end{bmatrix} \begin{bmatrix} \phi^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_c - \Delta t(1 - \theta) \mathbf{C}_c \phi^n + \mathbf{F}_c \end{bmatrix} \quad (2.55b)$$

where  $n$  is the actual timestep, and  $k$  if from the Krylov-Newton iteration scheme introduced in 2.3.2 .

---

**Algorithm 1** Solving the convection-diffusion equations with the SUPG

---

Assemble matrices  $\mathbf{M}_c$ ,  $\mathbf{C}_c$  and the vector  $\mathbf{F}_c$

Apply boundary conditions

**for**  $n = 1 \rightarrow N\_timesteps$  **do**

Solve 2.55b

update:  $\phi^n = \phi^{n-1}$

**end for**

---

**Algorithm 2** Solving the Navier-Stokes equations with the SUPG

---

**for**  $n = 1 \rightarrow N\_timesteps$  **do**

**for**  $k = 1 \rightarrow N\_nonlinear\_solver\_steps$  **do**

Assemble matrices  $\mathbf{M}$ ,  $\mathbf{K}$ ,  $\mathbf{G}$ ,  $\mathbf{G}'$  and vectors  $\mathbf{F}$ ,  $\mathbf{D}$

Apply boundary conditions

Solve 2.55a

**if**  $|\mathbf{v}^{k+1} - \mathbf{v}^k| < nonlinear\_tolerance$  **then**

update:  $\mathbf{v}^k = \mathbf{v}^{k+1}$

break;

**else**

update:  $\mathbf{v}^k = \mathbf{v}^{k+1}$

**end if**

**end for**

update:  $\mathbf{v}^n = \mathbf{v}^k$

**end for**

---

### 2.3.5 Mesh elements

In the finite element method, an approximate solution is sought in a finite dimensional function space. Seeking an approximate solution of a variational problem in a finite dimensional space means writing the solution as a linear combination of the elements of the basis that generates the space and then solve the algebraic linear equations systems obtained. An appropriate choice of basis is important for the performance of the numerical simulation. Smooth enough functions are obtained by belonging to spaces defined in 2.10 and 2.11. In the finite element method, the domain is divided into subregions in which the function for the basis of the solution space are defined. These subregions are called elements. The element shape should be such that the interpolation functions defined in it satisfy the requirement mentioned above. In incompressible problems such as the one stated in this chapter, there are numerical difficulties due to the constraint of the velocity field to satisfy the divergent free condition.

The LBB condition (formulated by Ladyzhenskaya, Babuška e Brezzi in the 1970's) establishes that the pressure and velocity spaces cannot be chosen arbitrarily; they must be compatible (4). In figure 3 we can see a couple of LBB-stable elements in two dimensions. Extension for three dimensions is straightforward.

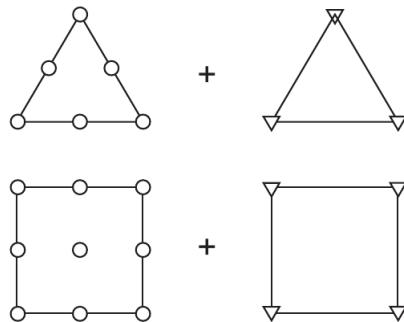


Figure 3 – Some LBB-stable elements: circles represent velocity nodes, and triangles pressure nodes

### 3 SEMI-LAGRANGIAN METHOD

#### 3.1 Introduction

The semi-Lagrangian method was firstly introduced to solve convection-diffusion systems aiming at two objectives: larger timesteps and stability. Also, high order elements lead to minimum diffusion errors. The semi-Lagrangian approximation has been used in meteorology for numerical weather forecasts, where big timesteps are essential to efficiency (6).

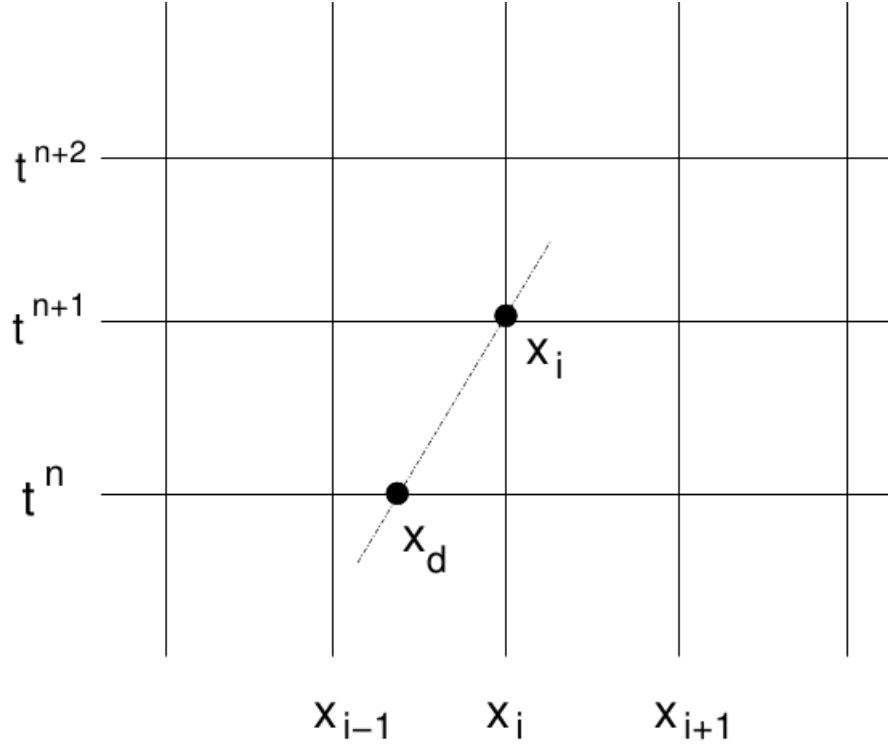
This method was introduced in the early 1980's by Robert (1980) and Pironneau (1981), and the main idea was the discretization of the Lagrangian time derivative instead of the Eulerian time derivative. We will explain the method on the convection-diffusion equation. Considering the material derivative of  $\phi$  we have (from equation 1.15):

$$\frac{D\phi}{Dt} - \frac{1}{Pe} \nabla^2 \sigma = f \quad (3.1)$$

Discretizing the last equation in the point  $x_i$  using an first order implicit scheme, we will have:

$$\frac{D\phi}{Dt} = \frac{\phi_i^{n+1} - \phi_d^n}{\Delta t} \quad (3.2)$$

The material derivative (in a exact solution) must be calculated along the *characteristic*, finding point  $x_d$  and solving the equation  $\frac{Dx}{Dt} = a$  backwards in time  $t^{n+1} \geq t \geq t^n$  using the initial condition:  $x(t^{n+1}) = x_i$ ; this is shown in figure 4. An integration method must be used in order to find the point's position in the previous timestep. If we use an first order scheme, then the approximate characteristic will be a straight line. Depending on that trajectory, three different situations can occur (12): the first two are shown by points 1 and 2, and the third by point 3 (figure 5). In the first trajectory, the point in the previous timestep was located next to the actual point, and inside problem domain. After identifying in which element the point has fallen, an interpolation is performed in order to know the value of  $\phi^n$ . In point 2 case,  $\phi^n$  is inside the problem's domain, but far from  $\phi^{n+1}$ . The difference between the first and second cases is the trajectory length. Using the same first order approximation (an straight line) used in point 1, the error will be bigger, because with little information (last and present iteration) we cannot calculate the real trajectory. In the third case, boundary conditions are used to interpolate  $\phi$ .

Figure 4 – Problem *characteristics*

### 3.2 Variational formulation

Taking into account the same notation, spaces, initial and boundary conditions of section 2.1 of chapter 2, the Galerkin formulation consists in finding solutions  $\mathbf{v}(\mathbf{x}, t) \in V_\Gamma$ ,  $p \in P$  such that:

$$\int_{\Omega} \left\{ \frac{D\phi}{Dt} + \nabla p - \frac{1}{Re} \nabla^2 \mathbf{v} - \frac{1}{Fr^2} \frac{\mathbf{g}}{g} \right\} w d\Omega = 0 \quad (3.3)$$

$$\int_{\Omega} [\nabla \cdot \mathbf{v}] q d\Omega = 0 \quad (3.4)$$

where  $w$  are the test functions. And for the convection-diffusion equation:

$$\int_{\Omega} \left\{ \frac{D\phi}{Dt} - \frac{1}{Pe} \nabla^2 \phi - f \right\} w d\Omega = 0 \quad (3.5)$$

Applying integration by parts (equations 2.20 and 2.21, and also equations 2.22 and 2.23) we have:

$$\int_{\Omega} \left( \frac{D\mathbf{v}}{Dt} + \nabla p \right) w d\Omega + \int_{\Omega} \frac{1}{Re} \nabla \mathbf{v} \nabla w d\Omega =$$

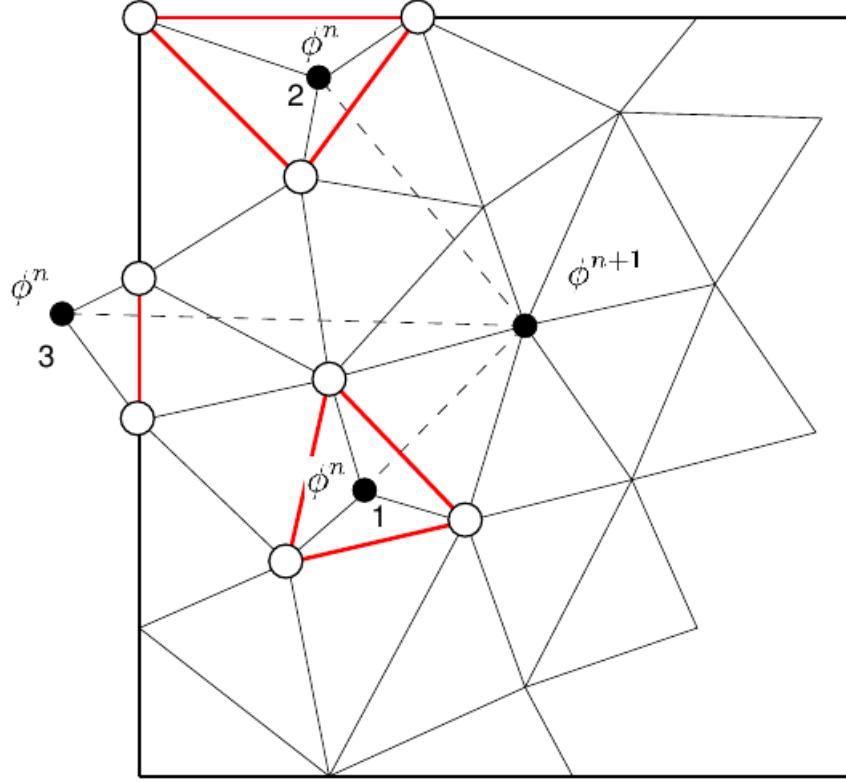


Figure 5 – Interpolation in the semi-Lagrangian method

$$\int_{\Omega} \frac{1}{Fr^2} \frac{\mathbf{g}}{g} w d\Omega + \int_{\Gamma_h} \frac{1}{Re} w h d\Gamma \quad (3.6)$$

$$\int_{\Omega} \frac{D\phi}{Dt} w d\Omega + \int_{\Omega} \frac{1}{Pe} \nabla \phi \cdot \nabla w d\Omega = \int_{\Omega} f w d\Omega + \int_{\Gamma_h} w h d\Gamma \quad (3.7)$$

### 3.3 Discretization and solution method

#### 3.3.1 Spatial discretization

Making the same domain decomposition as in 2.3.1 :

$$\begin{aligned} \int_{\Omega} \left( \frac{D\mathbf{v}_h}{Dt} + \nabla p_h \right) w_h d\Omega + \int_{\Omega} \frac{1}{Re} \nabla \mathbf{v}_h \nabla w_h d\Omega = \\ \int_{\Omega} \frac{1}{Fr^2} \frac{\mathbf{g}}{g} w_h d\Omega + \int_{\Gamma_h} \frac{1}{Re} w_h h d\Gamma \end{aligned} \quad (3.8)$$

$$\int_{\Omega} [\nabla \cdot \mathbf{v}_h] q_h d\Omega = 0 \quad (3.9)$$

$$\int_{\Omega} \frac{D\phi_h}{Dt} w_h d\Omega + \int_{\Omega} \frac{1}{Pe} \nabla \phi_h \cdot \nabla w_h d\Omega = \int_{\Omega} f w_h d\Omega + \int_{\Gamma_h} w_h h d\Gamma \quad (3.10)$$

with  $w_h \in V_0$ ,  $p \in P$  and  $c_h \in Y_0$ .

Now we can state the following approximation for  $w_h$ ,  $q_h$  and  $c_h$ :

$$w_h = \sum_{n=1}^{NV} \alpha_n w_n \quad (3.11)$$

$$q_h = \sum_{n=1}^{NP} \alpha_n q_n \quad (3.12)$$

$$c_h = \sum_{n=1}^{NC} \alpha_n w_n \quad (3.13)$$

Replacing 3.11 in 3.8:

$$\begin{aligned} & \int_{\Omega} \left( \frac{D\mathbf{v}_h}{Dt} + \nabla p_h \right) \left( \sum_{n=1}^{NV} \alpha_n w_n \right) d\Omega + \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v}_h \nabla \left( \sum_{n=1}^{NV} \alpha_n w_n \right) d\Omega \\ & - \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} \left( \sum_{n=1}^{NV} \alpha_n w_n \right) d\Omega - \frac{1}{Re} \int_{\Gamma_h} h \left( \sum_{n=1}^{NV} \alpha_n w_n \right) d\Gamma = 0 \end{aligned}$$

Rearranging terms:

$$\begin{aligned} & \alpha_1 \left\{ \int_{\Omega} \left( \frac{D\mathbf{v}_h}{Dt} + \nabla p_h \right) w_1 d\Omega + \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v}_h \nabla w_1 d\Omega - \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w_1 d\Omega - \frac{1}{Re} \int_{\Gamma_h} h w_1 d\Gamma \right\} \\ & + \alpha_2 \left\{ \int_{\Omega} \left( \frac{D\mathbf{v}_h}{Dt} + \nabla p_h \right) w_2 d\Omega + \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v}_h \nabla w_2 d\Omega - \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w_2 d\Omega - \frac{1}{Re} \int_{\Gamma_h} h w_2 d\Gamma \right\} \\ & \quad + \dots \\ & + \alpha_{NV} \left\{ \int_{\Omega} \left( \frac{D\mathbf{v}_h}{Dt} + \nabla p_h \right) w_{NV} d\Omega + \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v}_h \nabla w_{NV} d\Omega - \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w_{NV} d\Omega \right. \\ & \quad \left. - \frac{1}{Re} \int_{\Gamma_h} h w_{NV} d\Gamma \right\} = 0 \end{aligned}$$

This last expression must be valid for every  $w_h$ , hence the terms that multiply  $\alpha_1, \alpha_2 \dots \alpha_{NV}$  must be zero:

$$\begin{aligned}
\int_{\Omega} \left( \frac{D\mathbf{v}_h}{Dt} + \nabla p_h \right) w_1 d\Omega + \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v}_h \nabla w_1 d\Omega - \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w_1 d\Omega - \frac{1}{Re} \int_{\Gamma_h} h w_1 d\Gamma &= 0 \\
\int_{\Omega} \left( \frac{D\mathbf{v}_h}{Dt} + \nabla p_h \right) w_2 d\Omega + \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v}_h \nabla w_2 d\Omega - \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w_2 d\Omega - \frac{1}{Re} \int_{\Gamma_h} h w_2 d\Gamma &= 0 \\
&\vdots \\
\int_{\Omega} \left( \frac{D\mathbf{v}_h}{Dt} + \nabla p_h \right) w_{NV} d\Omega + \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v}_h \nabla w_{NV} d\Omega - \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w_{NV} d\Omega - \frac{1}{Re} \int_{\Gamma_h} h w_{NV} d\Gamma &= 0
\end{aligned}$$

Using the approximations 2.35 and 2.36 of  $\mathbf{v}_h$  and  $p_h$  we can express any of the  $NV$  equations before as:

$$\begin{aligned}
\int_{\Omega} \left( \frac{D(\sum_{n=1}^{NV} w_n(\mathbf{x}) \mathbf{v}_n(t))}{Dt} + \nabla(\sum_{n=1}^{NP} q_n(\mathbf{x}) p_n(t)) \right) w_j d\Omega + \frac{1}{Re} \int_{\Omega} \nabla(\sum_{n=1}^{NV} w_n(\mathbf{x}) \mathbf{v}_n(t)) \nabla w_j d\Omega \\
= \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w_j d\Omega + \frac{1}{Re} \int_{\Gamma_h} h w_j d\Gamma
\end{aligned}$$

where  $j = 1, \dots, NV$ . Rearranging some terms we obtain:

$$\begin{aligned}
\int_{\Omega} \left( \sum_{n=1}^{NV} w_n \frac{D\mathbf{v}_n}{Dt} + \sum_{n=1}^{NP} p_n \nabla q_n \right) w_j d\Omega + \sum_{n=1}^{NV} \frac{1}{Re} \int_{\Omega} \mathbf{v}_n \nabla w_n \nabla w_j d\Omega \\
= \frac{1}{Fr^2} \int_{\Omega} \frac{\mathbf{g}}{g} w_j d\Omega + \frac{1}{Re} \int_{\Gamma_h} h w_j d\Gamma
\end{aligned}$$

And now, considering the domain  $\Omega$  discretization into  $N_{el}$  elements:

$$\begin{aligned}
\sum_{l=1}^{N_{el}} \int_{\Omega_e} \left( \sum_{n=1}^{NV} w_n \frac{D\mathbf{v}_n}{Dt} + \sum_{n=1}^{NP} p_n \nabla q_n \right) w_j d\Omega + \sum_{l=1}^{N_{el}} \sum_{n=1}^{NV} \frac{1}{Re} \int_{\Omega_e} \mathbf{v}_n \nabla w_n \nabla w_j d\Omega \\
= \frac{1}{Fr^2} \sum_{l=1}^{N_{el}} \int_{\Omega_e} \frac{\mathbf{g}}{g} w_j d\Omega + \sum_{l=1}^{N_{el}} \frac{1}{Re} \int_{\Gamma_{h_e}} h w_j d\Gamma
\end{aligned}$$

With a similar process on the continuity and scalar transport equation we get:

$$\sum_{l=1}^{N_{el}} \sum_{n=1}^{NV} \int_{\Omega_e} (\nabla \cdot (w_n \mathbf{v}_n)) q_j d\Omega = 0$$

$$\begin{aligned} \sum_{l=1}^{N_{el}} \sum_{n=1}^{NC} \int_{\Omega_e} w_n \frac{D\phi_n}{Dt} d\Omega + \sum_{l=1}^{N_{el}} \sum_{n=1}^{NC} \int_{\Omega_e} \phi_n \frac{1}{Pe} \nabla w_n \cdot \nabla w_j d\Omega \\ = \sum_{l=1}^{N_{el}} \int_{\Omega_e} f w_j d\Omega + \sum_{l=1}^{N_{el}} \int_{\Gamma_{he}} w_j s d\Gamma \end{aligned}$$

which leads to the matrix form of the equations.

### 3.3.2 Matrix form of the equations

As can be seen from the last results obtained in the previous section, a *linear* system of equations is attained, due to the lack of the convective term in the Navier-Stokes equation:

$$\mathbf{M} \frac{D\mathbf{v}}{Dt} + \mathbf{K}\mathbf{v} - \mathbf{G}\mathbf{p} = \mathbf{F} \quad (3.14)$$

$$\mathbf{G}^T \mathbf{v} = \mathbf{D} \quad (3.15)$$

$$\mathbf{M}_c \frac{D\phi}{Dt} + \mathbf{C}_c \phi = \mathbf{F}_c \quad (3.16)$$

The solution vectors  $\mathbf{v}$ ,  $p$  and  $\phi$  are the ones defined in section 2.3.3 of chapter 2. All of these matrices can be shown in block form for better understanding of their structure:

$$\mathbf{M}_\rho = \begin{bmatrix} M_x & 0 & 0 \\ 0 & M_y & 0 \\ 0 & 0 & M_z \end{bmatrix}_{3NV \times 3NV} \quad (3.17a)$$

$$\mathbf{K} = \begin{bmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_z \end{bmatrix}_{3NV \times 3NV} \quad (3.17b)$$

$$\mathbf{F} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}_{NP \times 3NV} \quad (3.17c)$$

$$\mathbf{D} = \begin{bmatrix} D_x & D_y & D_z \end{bmatrix}_{NP \times 3NV} \quad (3.17d)$$

$$\mathbf{G} = \begin{bmatrix} G_x \\ G_y \\ G_z \end{bmatrix}_{3NV \times NP} \quad (3.17e)$$

$$\mathbf{G}^T = \begin{bmatrix} G_x & G_y & G_z \end{bmatrix}_{NP \times 3NV} \quad (3.17f)$$

$$\mathbf{M}_c = \begin{bmatrix} M_c \end{bmatrix}_{NC\theta \times NC\theta} \quad (3.17g)$$

$$\mathbf{C}_c = \begin{bmatrix} C_c \end{bmatrix}_{NC\theta \times NC\theta} \quad (3.17h)$$

$$\mathbf{F}_c = \begin{bmatrix} C_c \end{bmatrix}_{NC\theta \times 1} \quad (3.17i)$$

where the matrices terms are:

$$[M_{x_{ij}}] = [M_{y_{ij}}] = [M_{z_{ij}}] = \int_{\Omega_e} w_i w_j d\Omega \quad (3.18a)$$

$$[K_{x_{ij}}] = [K_{y_{ij}}] = [K_{z_{ij}}] = -\frac{1}{Re} \int_{\Omega_e} \nabla w_i \cdot \nabla w_j d\Omega \quad (3.18b)$$

$$[F_{x_i}] = [F_{y_i}] = [F_{z_i}] = \int_{\Omega_e} w_i \left( \frac{1}{Fr^2} \frac{\mathbf{g}}{g} \right) d\Omega \quad (3.18c)$$

$$[G_{x_{ij}}] = [G_{y_{ij}}] = [G_{z_{ij}}] = \int_{\Omega_e} q_j \nabla w_i d\Omega \quad (3.18d)$$

$$[M_{c_{ij}}] = \int_{\Omega_e} w_i w_j d\Omega \quad (3.18e)$$

$$[C_{c_{ij}}] = \frac{1}{Pe} \int_{\Omega_e} \nabla w_i \nabla w_j d\Omega \quad (3.18f)$$

$$[F_{c_i}] = \int_{\Omega_e} f w_i d\Omega \quad (3.18g)$$

### 3.3.3 Discretization in time

Using the semi-Lagrangian time discretization seen in section 3.1 , the matrix equations of section 3.3.2 become:

$$(\mathbf{M} + \Delta t \mathbf{K}) \mathbf{v}^{n+1} - \Delta t \mathbf{G} p^{n+1} = \mathbf{M} \mathbf{v}^n + \Delta t \mathbf{F} \quad (3.19a)$$

$$\mathbf{G}^T \mathbf{v}^{n+1} = \mathbf{D} \quad (3.19b)$$

$$(\mathbf{M}_c + \Delta t \mathbf{C}_c) \phi^{n+1} = \mathbf{M}_c \phi^n + \Delta t \mathbf{F}_c \quad (3.19c)$$

which can be written as (considering also equations 2.52):

$$\begin{bmatrix} \mathbf{M} + \Delta t \theta \mathbf{K} & -\Delta t \mathbf{G} \\ -\Delta t \mathbf{G}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}^{n+1} \\ \mathbf{p}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{M} \mathbf{v}_d^n + \Delta t \mathbf{F} - (1 - \theta) \Delta t \mathbf{K} \mathbf{v}^n \\ -\Delta t D \end{bmatrix} \quad (3.20a)$$

$$\begin{bmatrix} \mathbf{M}_c + \theta \Delta t \mathbf{C}_c \\ \mathbf{M}_c \phi_d^n + \Delta t F_c - (1 - \theta) \Delta t C_c \phi^n \end{bmatrix} = \begin{bmatrix} \phi^{n+1} \end{bmatrix} \quad (3.20b)$$

It has to be noticed that the matrices are symmetric. This coupled system of equations is solved once on each timestep after having solved the semi-Lagrangian problem.

---

#### Algorithm 3 Solving the scalar transport equation with the semi-Lagrangian method

Assemble matrices  $\mathbf{M}_c$ ,  $\mathbf{C}_c$  and the vector  $\mathbf{F}_c$

Apply boundary conditions

**for**  $n = 1 \rightarrow N\_timesteps$  **do**

    With  $\phi^{n-1}$  compute  $\phi_d^n$  according to section 3.1

    Compute  $\phi^n$  by solving 3.20b

**end for**

---



---

#### Algorithm 4 Solving the Navier-Stokes equations with the semi-Lagrangian method

**for**  $n = 1 \rightarrow N\_timesteps$  **do**

    Assemble matrices  $\mathbf{M}$ ,  $\mathbf{K}$ ,  $\mathbf{G}$  and vectors  $\mathbf{F}$ ,  $\mathbf{D}$

    Apply boundary conditions

    With  $\mathbf{v}^{n-1}$ , compute  $\mathbf{v}_d^n$  according to section 3.1

    Compute  $\mathbf{v}^n$  by solving 3.20a

**end for**

---

## 4 NUMERICAL TESTS

### 4.1 Introduction

A certain number of test were run in order to study the behavior of the semi-Lagrangian method. To validate the results we compare them to results found in literature, and in order to have available a full solution with a method that renders good results, we solve the same problems with the SUPG. The code for this tests was written using the *libMesh* parallel adaptive finite element library (13) from the CFDlab of the University of Texas at Austin. All computations use the PETSc library (14) from Argonne National Laboratory to solve the linear systems. The 2D meshes employed were generated with the Triangle library by Jonathan Richard Shewchuk (15) or with the mesh generation tools from the *libMesh* library.

First we will present more information on the libraries and the specific implementation. Then some brief tests run on the convection-diffusion equation, focused on spurious oscillations generation. Third, the tests ran on the Navier-Stokes equations: the lid-driven cavity, which is a very well known benchmark with much results available in literature, and the backward facing step problem, another benchmark with very interesting physics. These applications will focus in the numerical aspects of the semi-Lagrangian finite element algorithm.

### 4.2 The *libMesh* library

*libMesh* is a project developed in the CFDlab at the University of Texas at Austin since March 2002. It provides support for numerical simulation of partial differential equations using arbitrary unstructured discretizations on serial and parallel platforms, in one, two and three dimension steady and transient simulations employing various finite element types. It also provides Adaptive Mesh Refinement (AMR) (16).

This library makes use of highly optimized existing software, as PETSc (Portable, Extensible Toolkit for Scientific Computation) (14) for solving the linear systems for both serial and parallel platforms, MPI (Message Passing Interface) for PETSc functionality and the *libMesh* parallel implementations (17), SLEPc (Scalable Library for Eigenvalue Computations) for sparse matrices eigenvalue calculations in parallel (18), BLAS (Basic Linear Algebra Subprograms) for basic vector and matrix operations (19), LAPACK for PETSc internal functionality (20), and LASPack for serial solution of linear systems (21). For mesh generation, *libMesh* uses Triangle (15) and Tetgen (22) to generate triangular and tetrahedral meshes in 2D and 3D respectively. All of them were compiled in optimized mode for better performance.

Parallelism in *libMesh* is present at the matrix assembly and linear algebra levels (solving the linear systems). Parallel architecture paradigm provides one strong benefit: scalability. Pre processing (reading/writing a mesh and input data, partitioning and distribution) is still serial on *libMesh*. Non overlapping domain decomposition approach is used in *libMesh*, each subdomain is assigned to an individual processor. Domain decomposition aims to balance the load on each processor and the communication between processors. Mesh partition is provided by METIS (23) for coarse parallel granularity when the number of partitions is less or equal to eight, and with the *k*-way scheme otherwise (23).

In this work, all parallel simulations were ran on a node with two Intel Quad-Core Xeon E5620, with 16Gb of RAM available (DDR3-1066), with 8 processes (one per core). All linear systems were solved using GMRES (Generalized minimal residual method) (24) with restart after 30 iterations. The preconditioner is of Jacobi-type where each processor sub-block uses an overlapping additive Schwartz method with an incomplete lower-upper factorization at the sub-block level with no fill (ILU-0). Spatial integration is performed with Gauss quadrature rules sufficient to integrate third order polynomials exactly.

### 4.3 Tests on the Convection-Diffusion equation

As was introduced before, the analysis of the convection-diffusion equation (also called scalar transport equation) is important because of its several applications along with the Navier-Stokes equations ((6), (12), (25)). Spurious oscillations are generated if the Galerkin method is used, specially when these quantities leave the mesh while been carried by the fluid, which is an important physical situation and a common one. We will study in these tests the amplitude of these oscillations generated by both methods: SUPG and semi-Lagrangian.

#### 4.3.1 The rotating cone test

This is a standard problem for advection algorithms (2). The exact solution consist of a rigid rotation of the cone about the center of the domain. In figure 6 we can see a scheme of the problem statement. A mesh of  $30 \times 30$  nine nodes quadrilateral elements is selected, over a domain  $[-1, 1] \times [-1, 1]$ ; and  $\Delta t = 0.025$ , with 250 time steps per full rotation of the cone. Here, we have taken a diffusivity equal to zero, and therefore  $Pe = \infty$  (pure advection).

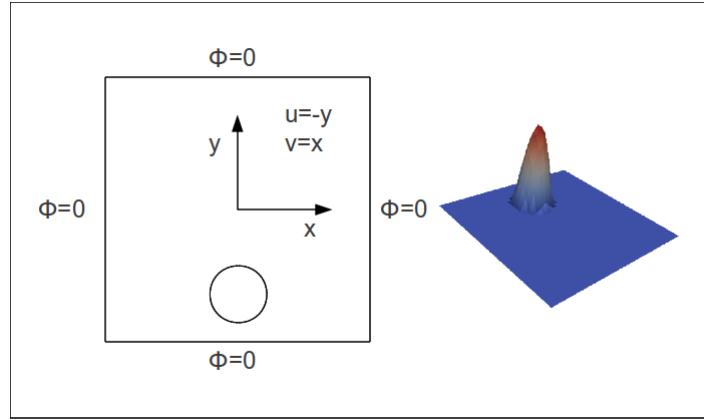


Figure 6 – Rotating cone problem statement

#### 4.3.2 Rotating cone test results

Results for a full rotation are shown in figure 7. The Galerkin approach presented no diffusion error, but trailing waves with a maximum size of 8,21% of the initial cone height appear all over the domain. With the SUPG stabilization there are no trailing waves across the domain, just a little depression behind the cone, of about 4.4% of the cone's initial height, but the method presents diffusion: cone height after one rotation is about 88% of initial size. The semi-Lagrangian approximation shows the same behavior as the SUPG method, but less diffusion: cone height is 94,4% of initial, after a full rotation.

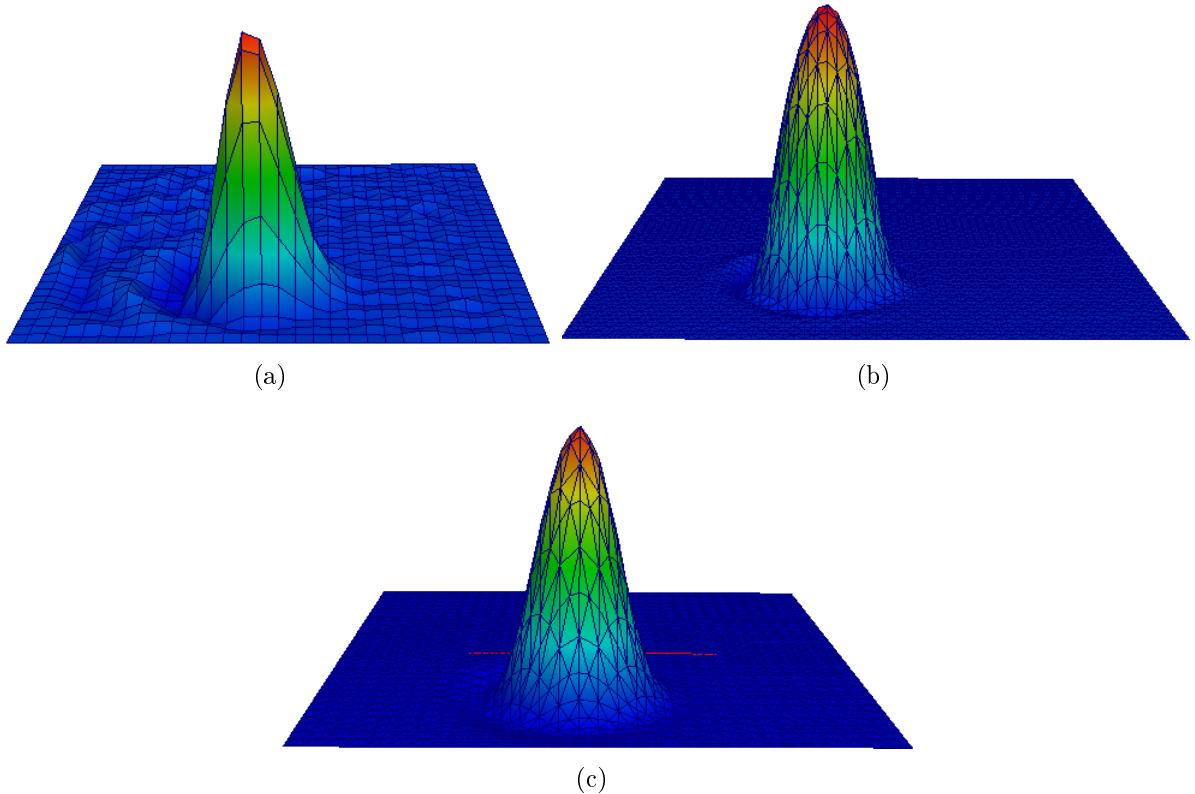


Figure 7 – Results after a full rotation: (a) Galerkin; (b) SUPG (c) semi-Lagrangian

The semi-Lagrangian is implicitly an upwind method, and hence it eliminates the spurious oscillations. Another result of interest is that with second order interpolation, the semi-Lagrangian shows less diffusive results than the SUPG. Lower order interpolation leads to very diffusive results, as we will see in the following tests.

#### 4.3.3 The cone impinging a mesh boundary test

Another test of interest is the cone impinging a mesh boundary. In transport problems is often necessary for certain quantities to leave the mesh while been carried by the fluid. A mesh of  $20 \times 100$  nine node quadrilateral elements was used in a  $[0.0, 5.0] \times [0.0, 1.0]$  domain. A Crank-Nicholson time advancing scheme was implemented in each of the three methods. Neumann boundary conditions were implemented (zero on every boundary). The initial condition is a cosine hill of width 0.5 in  $(0.5, 0.5)$ , and height 1.0. The fluid velocity is constant and equal to  $(1, 0)$ .

#### 4.3.4 Cone impinging a mesh boundary test results

Galerkin leaves trailing waves all over the domain, and when the cone has recently left the mesh, oscillations with a maximum height of 7.5% of the initial cone height remains. SUPG leaves minimum oscillations that exits the mesh rapidly, just behind the cone, with a maximum height of 7.1%. With the semi-Lagrangian method, the cone exits the mesh cleanly, with no oscillations, as can be seen in figure 8.

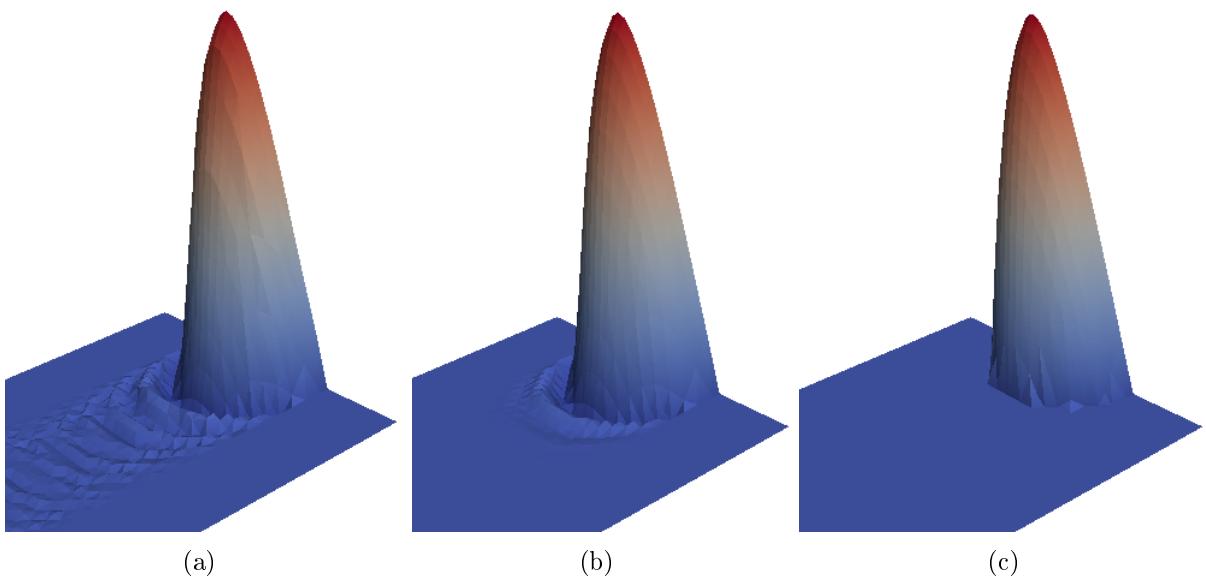


Figure 8 – Results after a full rotation: (a) Galerkin; (b) SUPG (c) semi-Lagrangian

In the previous test quadratic elements were used. It has been noticed in previous works with the semi-Lagrangian ((12), (25), (6)) that using lower order interpolation (even with fine meshes) gives very diffusive results. This can be seen in figure 9. This problem can be solved refining the mesh, as seen in figure 10. Here we have taken a mesh with the double of elements to compensate the lower order elements. But on the same mesh, with low order interpolation, the SUPG shows better results. We will return to this in section 4.4.4 .

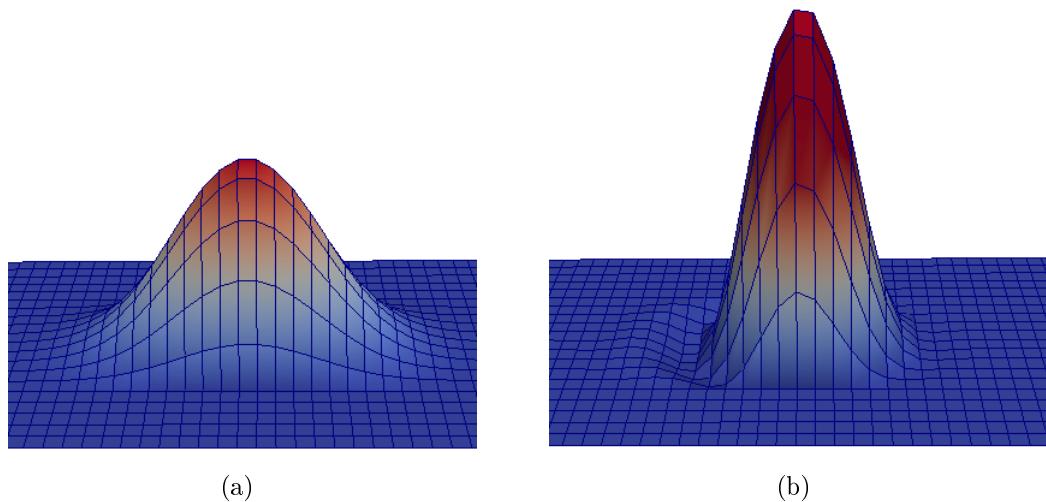


Figure 9 – Results on the cone impinging a mesh boundary (4 node quadrilaterals instead of 9). (a): semi-Lagrangian; (b) SUPG.

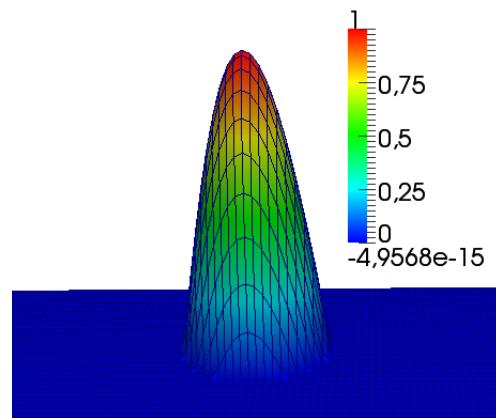


Figure 10 – semi-Lagrangian results with 4 node quadrilaterals and a finer mesh

## 4.4 Tests on the Navier-Stokes equations

#### 4.4.1 The 2D lid-driven cavity test

In order to evaluate the characteristics of the semi-Lagrangian method and to test its feasibility in the context of high Reynolds numbers we analyze the lid-driven cavity problem, a well-known benchmark for N-S solvers. This flow is not only technologically important, it is of great scientific interest because it displays almost all fluid mechanical phenomena in the simplest of geometrical settings. Thus corner eddies, longitudinal vortices, nonuniqueness, transition and turbulence all occur naturally and can be studied in the same closed geometry. This facilitates the comparison of results

from experiment, analysis, and computation over the whole range of Reynolds numbers (26).

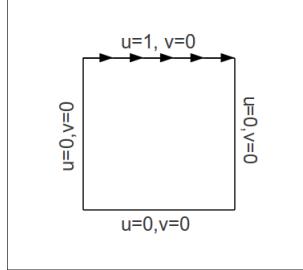


Figure 11 – Lid-driven cavity problem statement

The problem statement is shown in figure 11. It is a square cavity with no-slip boundary conditions, and velocity in the upper boundary is equal to 1 in the  $x$  direction. Pressure is pinned to zero in the lower left corner.

Erktuk et al. (27) solved the same problem using an streamfunction and vorticity formulation using finite differences in a very fine mesh of  $601 \times 601$  elements. Hachem et al. (1) used an multiscale method for solving the same problem, over two meshes: a fine one of  $180 \times 180$  elements, and an coarse one of  $64 \times 64$  elements. Ghia et al. (28) applied a second-order accurate finite difference method using a fine grid of  $257 \times 257$ . The work of Hachem has shown very good results compared with (27), even with a coarser mesh. We will refer mostly to the works of Hachem and Erkturk.

The meshes employed in our simulations are presented in figures 12 and 13. They are refined next to the boundary for better resolution of the boundary layer. This mesh is composed of triangular elements with quadratic interpolation for velocity, and linear interpolation for pressure (to satisfy the Babuska-Brezzi condition). Reynolds numbers of 1.000, 5.000 and 10.000 are established. For all of them, calculations are performed usings increasing CFLs of 1.1375, 1.52, 2.275 and 4.55, in order to compare the semi-Lagrangian versus the SUPG. Results are compared with those of (27) and (1).

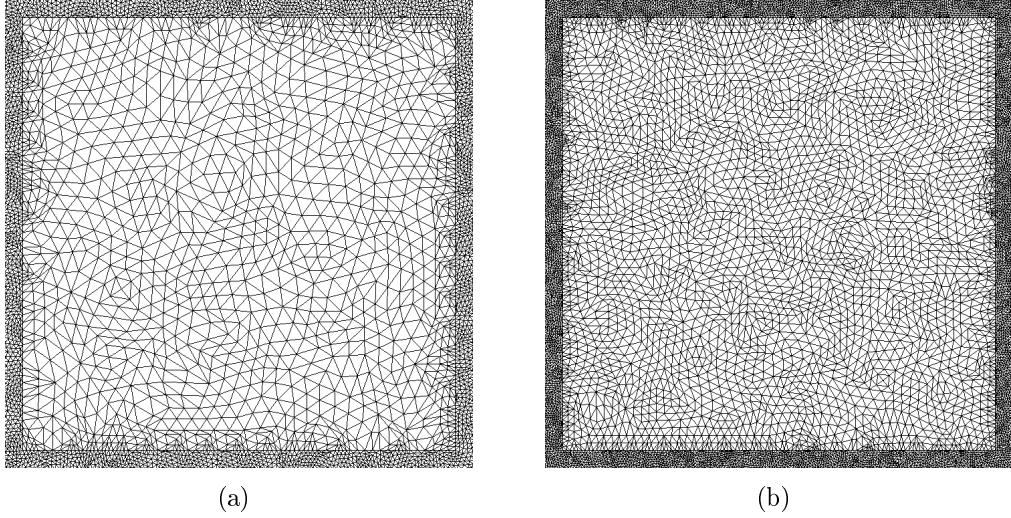


Figure 12 – Lid driven cavity 2D meshes: (a) 2.000 elements. (b) 5.000 elements.

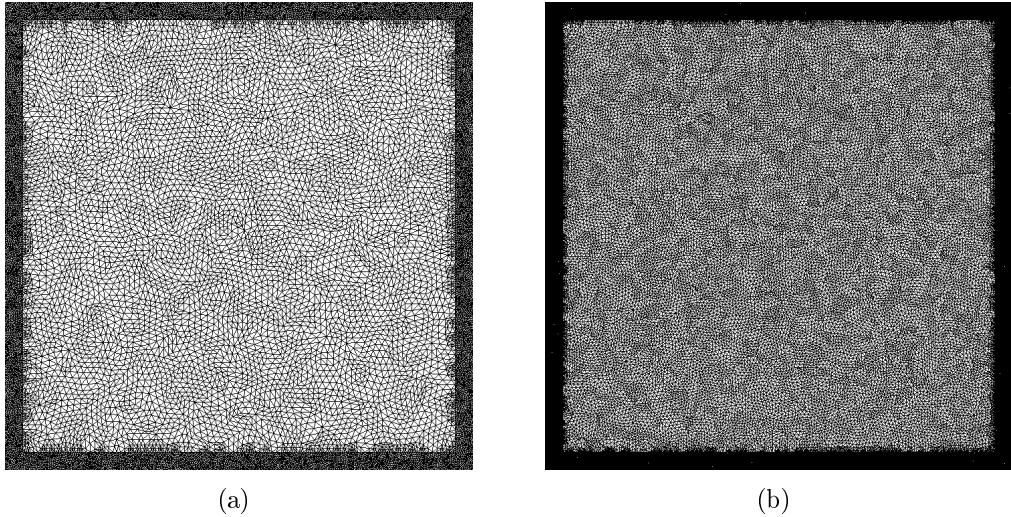


Figure 13 – Lid driven cavity 2D meshes: (a) 10.000 elements. (b) 40.000 elements.

We consider that steady state has been reached once the L2-norm of the normalized velocity difference between two timesteps is lower than  $1.0e - 4$ . Velocity profiles, position of vortex, streamline plots and graphics of error of the solutions (compared with a more accurate solution) are analysed, following the work of Hachem (1) and the verification and validation procedures seen in (29). The error of a certain simulation is computed as:

$$err(h) = \left( \sum_{x,y} (\mathbf{v}_{ref}^i - \mathbf{v}_k^i)^2 \right)^{\frac{1}{2}} \quad (4.1)$$

where  $h$  is making reference to a certain mesh and  $\mathbf{v}_{ref}$  is the velocity at the points of the reference mesh (the accurate solution). Comparison is made against the SUPG results computed with the finest mesh (40.000 elements).

#### 4.4.2 2D lid-driven cavity test

Velocity profiles are shown in figures 15, 16 and 17. In those figures we expose extreme cases of resolution: our finer and our coarser mesh. Our results in the meshes with intermediate resolution (5.000 and 10.000 elements) have shown an intermediate adequacy (as expected) between the coarsest and the finest mesh results, and therefore are not exhibited. The very accurate solution from Erkturk (27) and our SUPG solution are plotted as well as the semi-Lagrangian method for several CFLs (equation 2.34) in order to show the influence of taking bigger time steps. As can be

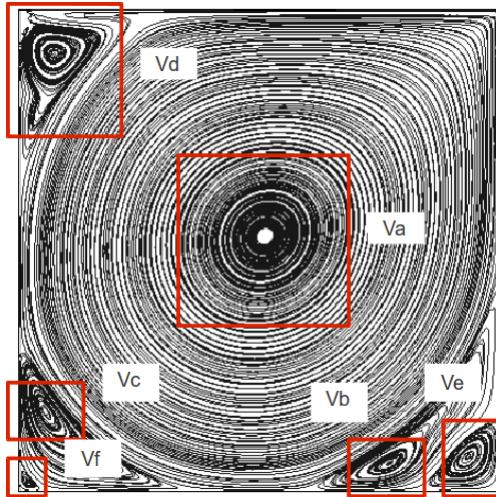


Figure 14 – Position of vortex until  $Re=10.000$

seen, for low CFLs ( $CFL \simeq 1.0$ ), semi-Lagrangian approximation is good, very close to the SUPG solution even for the highest Reynolds solution computed, however, the SUPG solution is slightly better comparing to (27). As the CFL grows, approximation becomes less accurate; this effect is more notorious when Reynolds number is higher; in this case  $CFL \geq 3.0$  grants poor approximations. As the only parameter that is being changed in this test (for the same mesh and same Reynolds number) is  $\Delta t$  then we can see that the origin of the problem stated before is strongly associated to the linear trajectory used to approximate the material derivative in 3.1 .

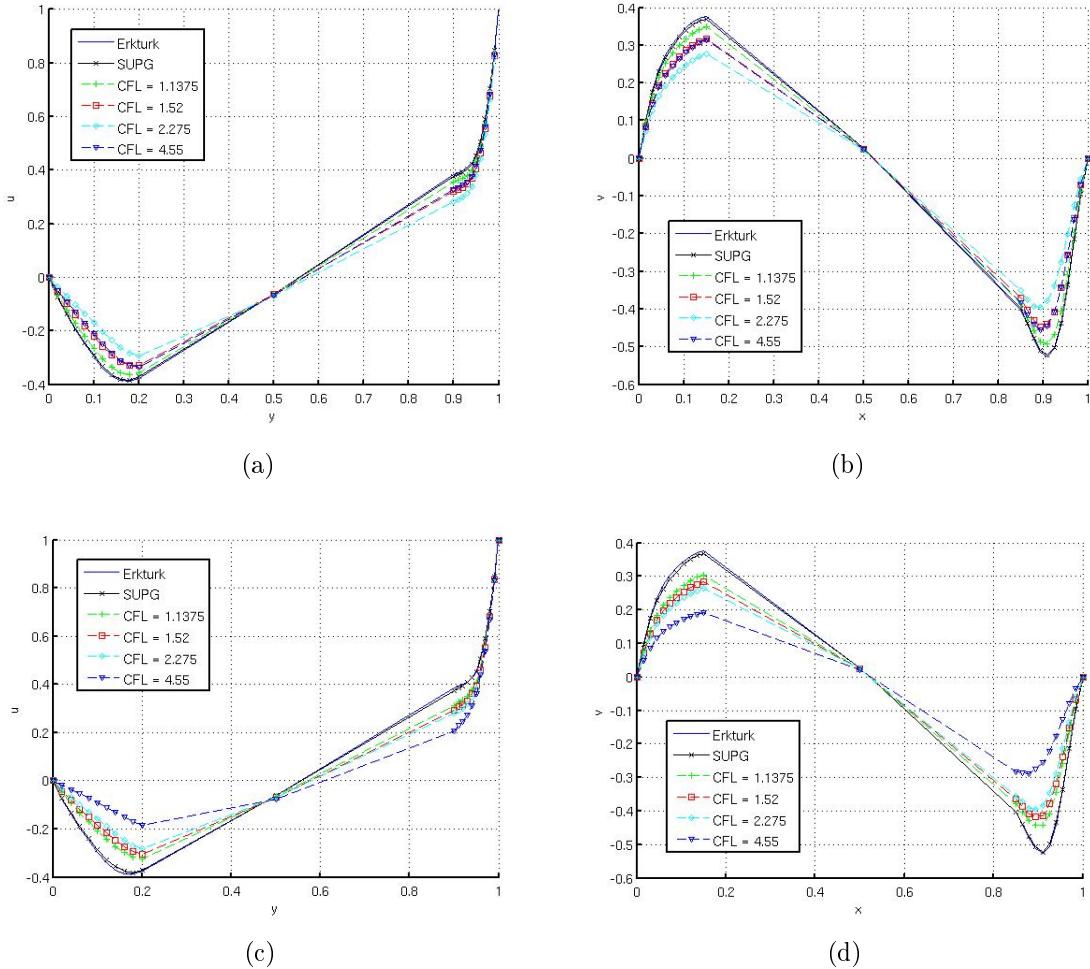


Figure 15 – Velocity profiles at  $x = 0.5$  (left) and  $y = 0.5$  (right) ( $\text{Re} = 1.000$ ).  
(a) and (b): mesh of 40.000 elements; (c) and (d): mesh of 2.000 elements.

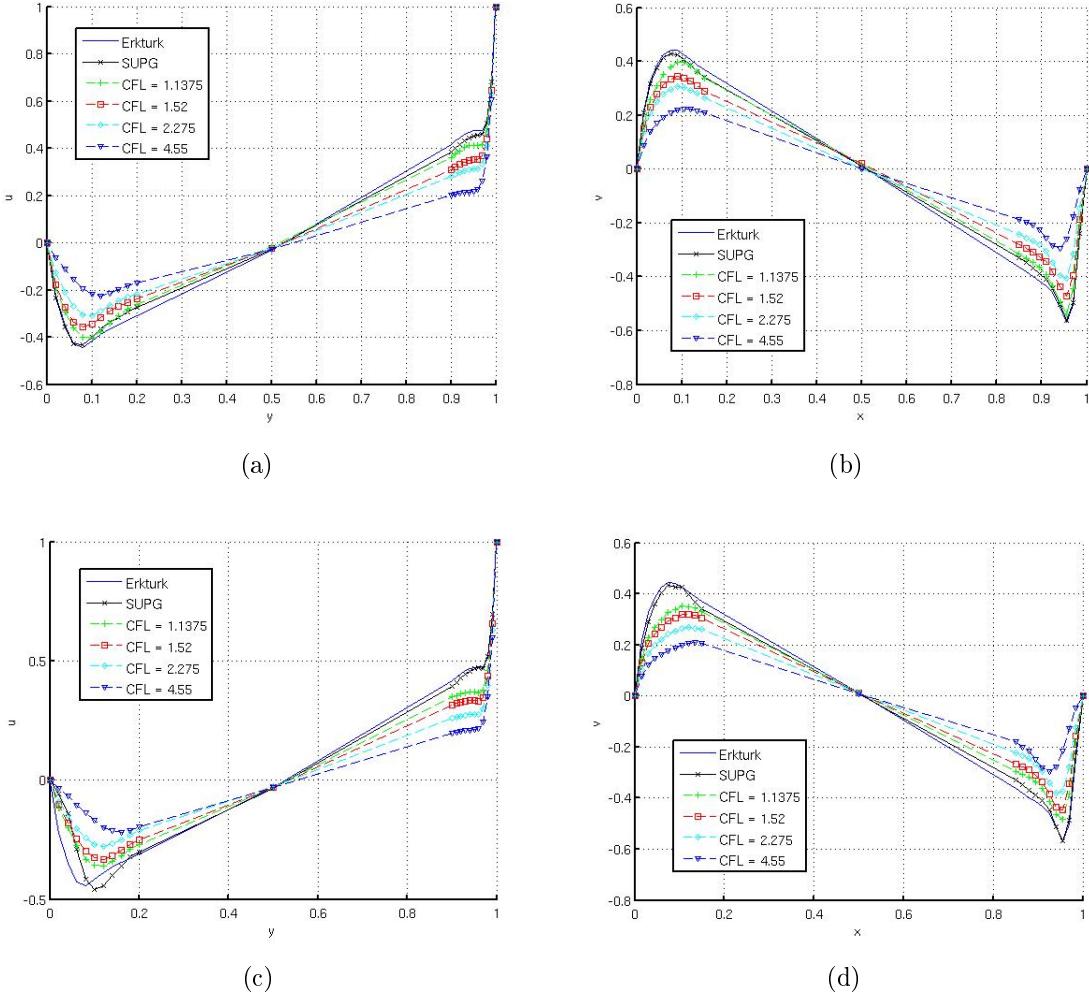


Figure 16 – Velocity profiles at  $x = 0.5$  (left) and  $y = 0.5$  (right) ( $Re = 5.000$ ). (a) and (b): mesh of 40.000 elements; (c) and (d): mesh of 2.000 elements.

Streamline plots are helpful to see vortex resolution. Here to compare our results we have (27) and (1). In figure 18 we have plotted the streamlines of the SUPG solution for the three Reynolds numbers, and in 19, 20 and 21 the semi-Lagrangian ones; all of them for the 40.000 elements mesh.

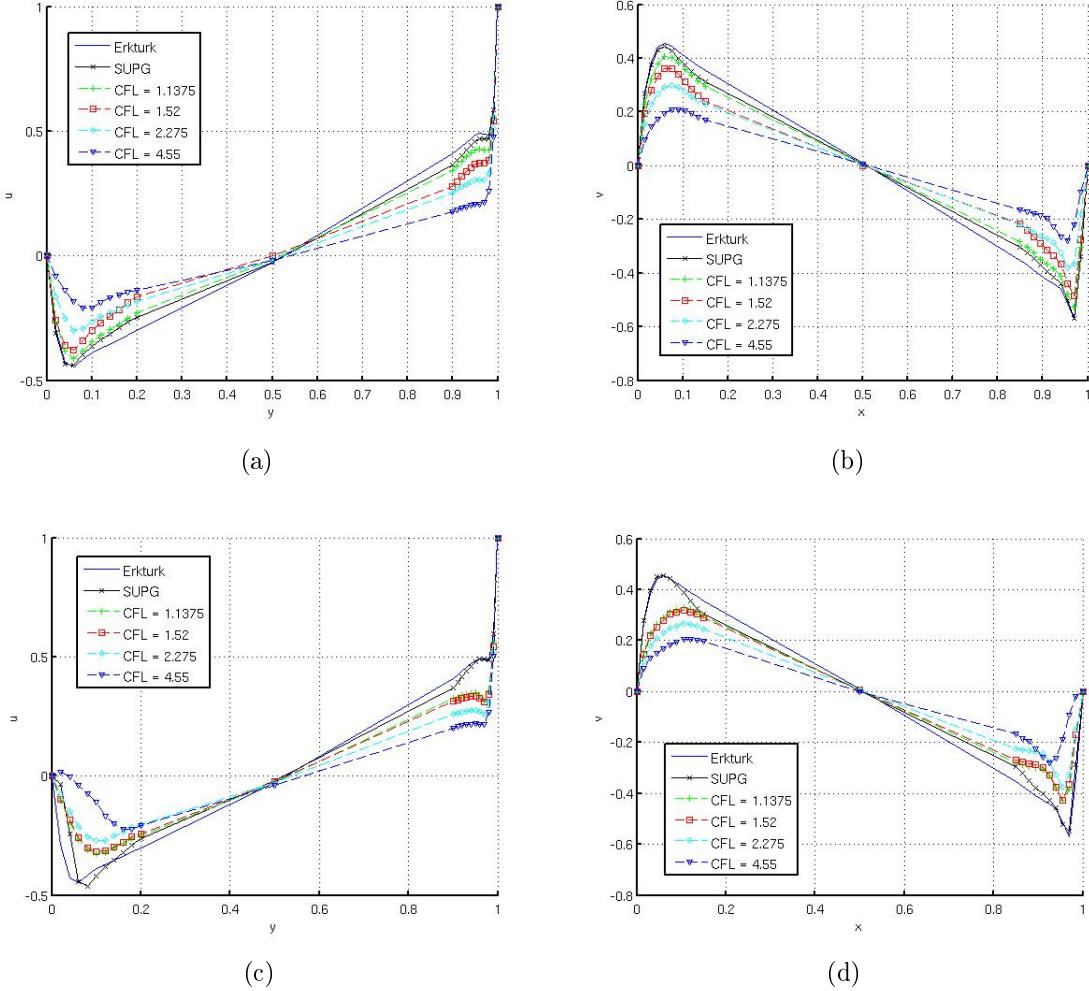


Figure 17 – Velocity profiles at  $x = 0.5$  (left) and  $y = 0.5$  (right) ( $Re = 10.000$ ). (a) and (b): mesh of 40.000 elements; (c) and (d): mesh of 2.000 elements.

Some important conclusions can be drawn upon the semi-Lagrangian. It can be seen that for lower Reynolds the solution is not very sensible to CFL. Vortex formation and position (see table 2) vary very little( $Re=1.000$ ); it only gets more diffusive and tends to underpredict the fluid velocity (see for example figures 4.15(a) and 4.15(b)).

Table 1 – Number of resolved vortices in function of Reynolds numbers

Reynolds	Vortices
1.000	3
5.000	5
10.000	6

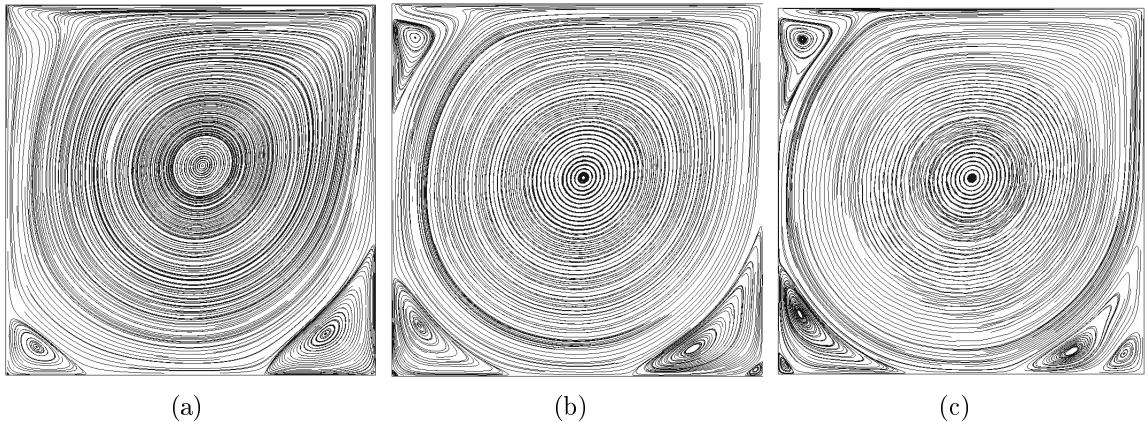


Figure 18 – SUPG streamlines for: (a)  $\text{Re} = 1.000$ ; (b)  $\text{Re} = 5.000$ ; (c)  $\text{Re} = 10.000$ ;

As Reynolds increases, CFL influence in the solution becomes more important. This can be noticed looking at figures 20, 21 and table 2. The semi-Lagrangian cannot resolve the  $V_e$  and  $V_f$  vortices (see figure 14) for higher Reynolds and high CFL.

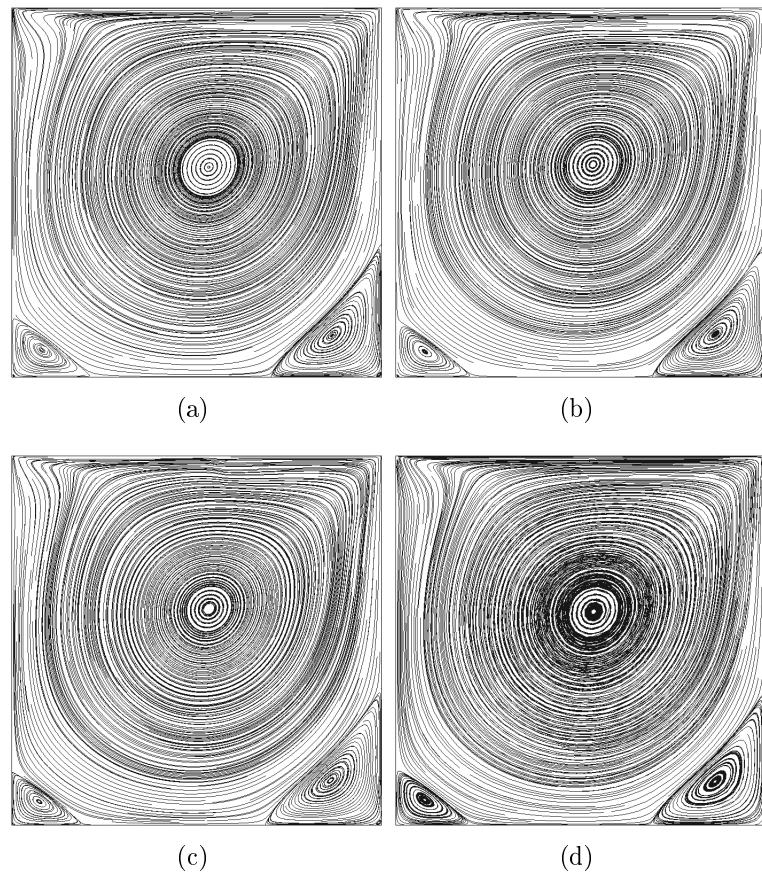


Figure 19 – Semi-Lagrangian streamlines for  $\text{Re}=1.000$ . CFL: (a) 1.1375; (b) 1.52; (c) 2.275; (c) 4.55.

For the SUPG solution an implicit esqueme (backward Euler) was used, with a larger timestep ( $\Delta t = 0.1$ ,  $CFL \approx 18.2$ ) and for the semi-Lagrangian we employed a Crank-Nicolson scheme, which is second order in time. It seems that the main source of error is in the linear trajectory used to estimate the material derivative. For higher Reynolds and CFLs a higher order interpolation for the particles trajectories could lead to improved results. The graphics of error of the computed solutions versus the CFL number (figure 22) reflect what has been discussed.

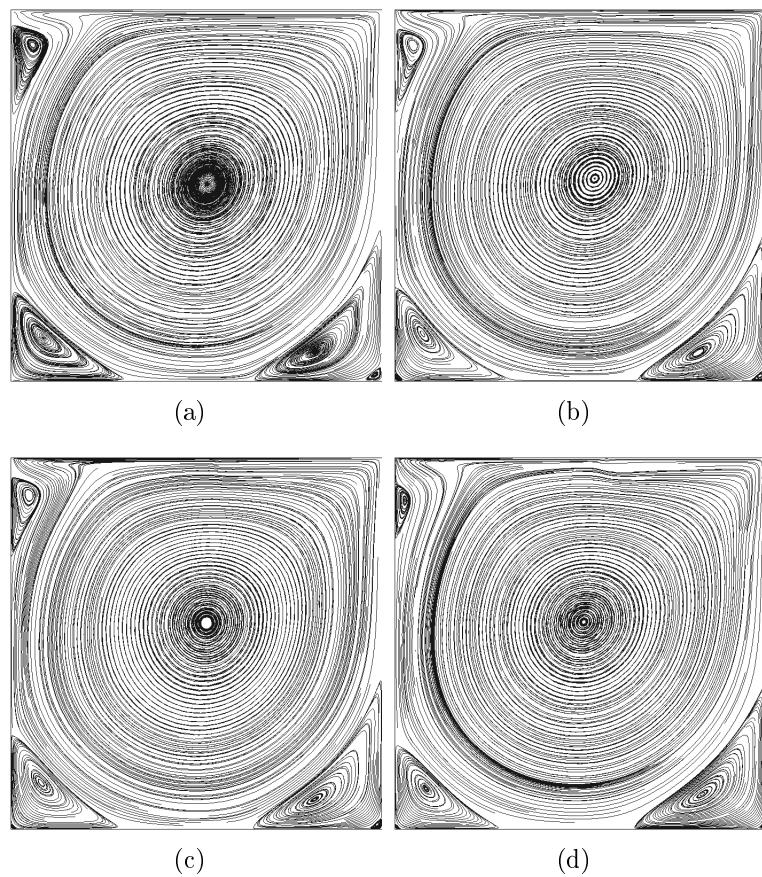


Figure 20 – Semi-Lagrangian streamlines for  $Re=5.000$ . CFL: (a) 1.1375; (b) 1.52; (c) 2.275; (c) 4.55.

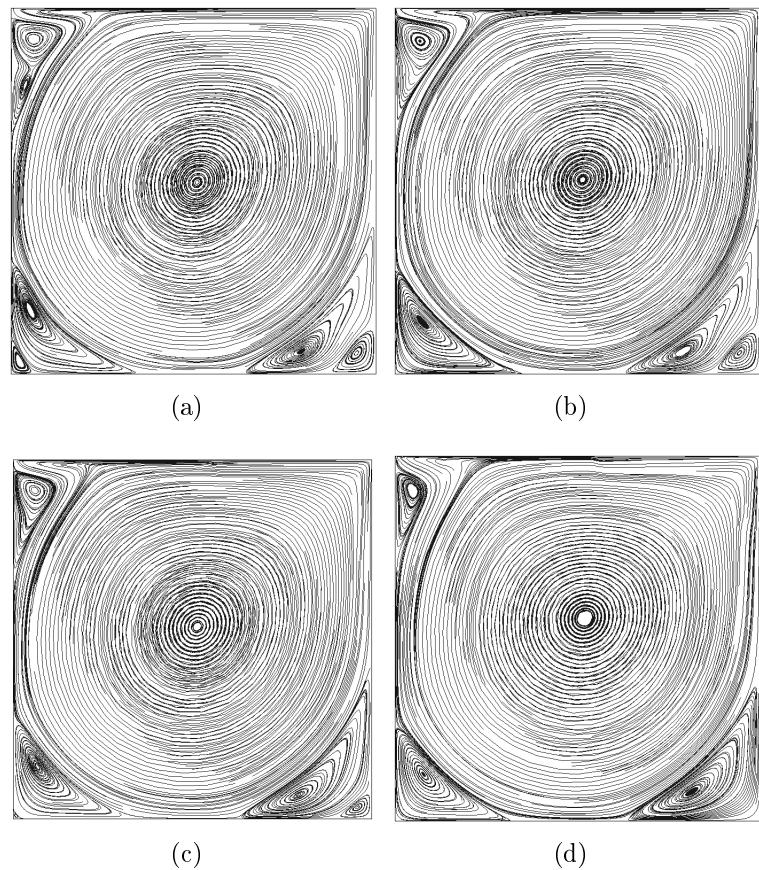


Figure 21 – Semi-Lagrangian streamlines for  $\text{Re}=10.000$ . CFL: (a) 1.1375; (b) 1.52; (c) 2.275; (c) 4.55.

Table 2 – Vortex coordinates

Vortex	Reynolds	1.000	5.000	10.000
Va	Erkturk et al.	(0.53000, 0.56500)	(0.51500, 0.53500)	(0.51170, 0.53000)
	SUPG	(0.53158, 0.56527)	(0.51580, 0.53288)	(0.53001, 0.53554)
	SL CFL=1.1375	(0.53206, 0.56839)	(0.52851, 0.53072)	(0.53051, 0.53210)
	SL CFL=1.52	(0.53268, 0.57617)	(0.53869, 0.54553)	(0.51941, 0.49617)
	SL CLF=2.275	(0.53290, 0.58594)	(0.52699, 0.55514)	(0.51327, 0.53634)
	SL CLF=4.55	(0.53439, 0.57800)	(0.50916, 0.55785)	(0.52044, 0.55617)
Vb	Erkturk et al.	(0.86330, 0.11170)	(0.80500, 0.07330)	(0.77670, 0.06000)
	SUPG	(0.86473, 0.11215)	(0.80957, 0.07202)	(0.80172, 0.06224)
	SL CFL=1.1375	(0.86562, 0.11332)	(0.82024, 0.07929)	(0.80231, 0.06176)
	SL CFL=1.52	(0.86378, 0.11690)	(0.81741, 0.07624)	(0.80418, 0.06409)
	SL CLF=2.275	(0.86149, 0.12168)	(0.81830, 0.07991)	(0.79616, 0.06717)
	SL CLF=4.55	(0.86316, 0.11870)	(0.82860, 0.09441)	(0.81260, 0.07965)
Vc	Erkturk et al.	(0.08330, 0.07830)	(0.07330, 0.13670)	(0.05830, 0.16330)
	SUPG	(0.08271, 0.07679)	(0.07303, 0.13712)	(0.05937, 0.16480)
	SL CFL=1.1375	(0.08137, 0.07466)	(0.08710, 0.11645)	(0.06100, 0.16210)
	SL CFL=1.52	(0.07773, 0.06981)	(0.07425, 0.12744)	(0.06203, 0.15982)
	SL CLF=2.275	(0.07347, 0.06446)	(0.08106, 0.12479)	(0.06790, 0.14552)
	SL CLF=4.55	(0.07747, 0.06908)	(0.08313, 0.10967)	(0.07789, 0.12812)
Vd	Erkturk et al.		(0.06330, 0.91000)	(0.07170, 0.91170)
	SUPG		(0.06247, 0.90966)	(0.06648, 0.91382)
	SL CFL=1.1375		(0.05906, 0.90604)	(0.06320, 0.92037)
	SL CFL=1.52		(0.04914, 0.90560)	(0.05939, 0.91448)
	SL CLF=2.275		(0.04839, 0.90015)	(0.06031, 0.91590)
	SL CLF=4.55		(0.02745, 0.88626)	(0.04741, 0.90678)
Ve	Erkturk et al.		(0.97830, 0.01830)	(0.93500, 0.06670)
	SUPG		(0.97963, 0.01806)	(0.94585, 0.05576)
	SL CFL=1.1375		(0.98016, 0.01717)	(0.94870, 0.05891)
	SL CFL=1.52		(0.98424, 0.01436)	(0.95290, 0.04398)
	SL CLF=2.275		(0.98680, 0.01238)	(0.95987, 0.03290)
	SL CLF=4.55		(0.99154, 0.00938)	not resolved
Vf	Erkturk et al.			(0.01670, 0.02000)
	SUPG			(0.02002, 0.02460)
	SL CFL=1.1375			(0.01560, 0.01879)
	SL CFL=1.52			(0.01487, 0.01839)
	SL CLF=2.275			(0.00865, 0.00919)
	SL CLF=4.55			not resolved

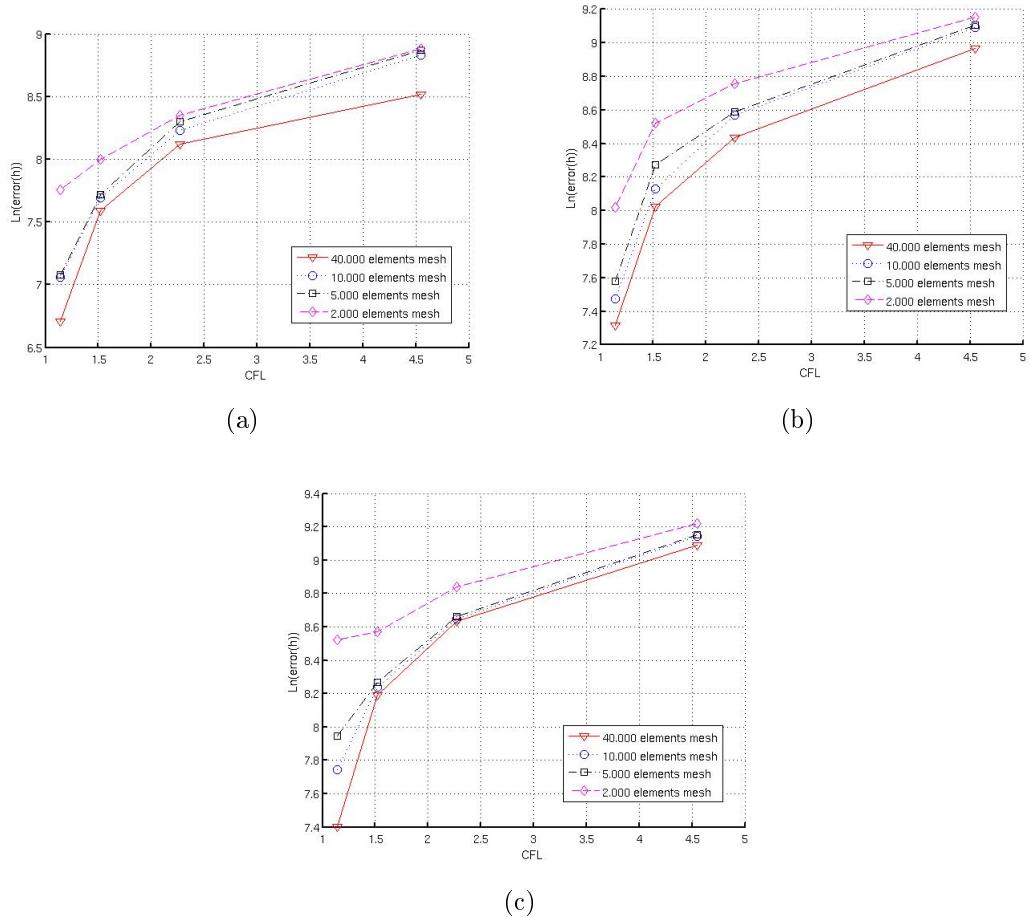


Figure 22 – Error: (a)  $Re=1.000$ ; (b)  $Re=5.000$ ; (c)  $Re=10.000$ ;

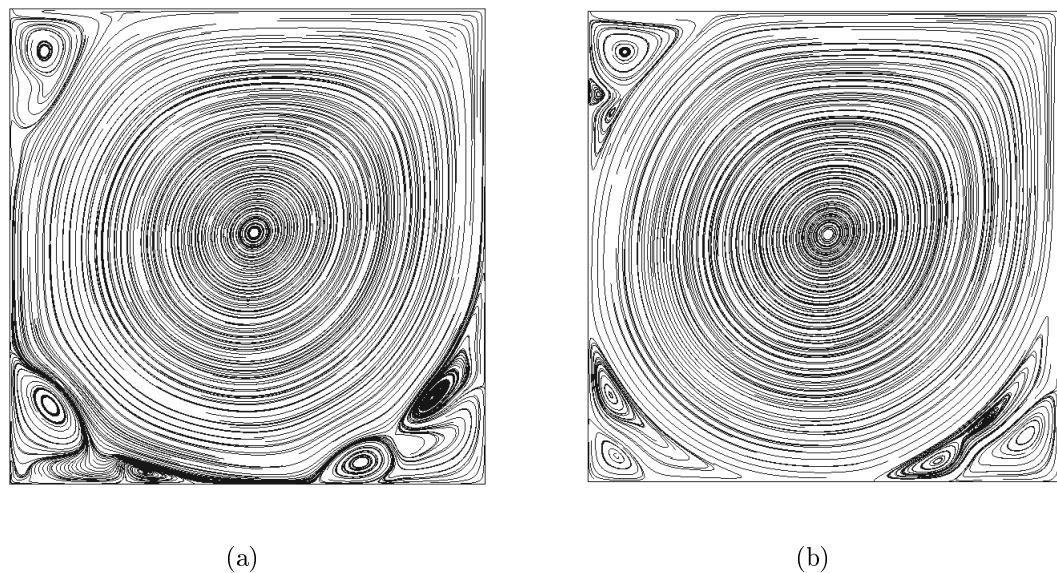


Figure 23 – Results for  $Reynolds=20.000$ : (a) semi-Lagrangian; (b) SUPG.

If the Reynolds number is furtherly increased the semi-Lagrangian solution presents oscillations (and even instability for very low CFLs), as can be seen in figure 23 ( $Re = 20.000$ ). Here, the semi-Lagrangian solution was computed with a  $CFL \simeq 1$  and the SUPG with a  $\Delta t = 0.1$  ( $CFL \simeq 18.2$ ), and both of them with the 40.000 elements mesh. The SUPG solution fits the results shown in (1). This problem will be analysed with more attention in the next section.

#### 4.4.3 The backward facing step test

This problem has been object of detailed experimental and numerical studies. With the lid-driven cavity, are two of the most popular problems for testing fluid flow algorithms in 2D. Cruchaga solved the problem using a streamline operator technique (30) with Reynolds up to 6.600. Erkturk (31) used a very efficient finite difference scheme that proved to be stable even with high Reynolds numbers. Lee and Mateescu (32) made an experimental investigation for Reynold numbers until 3.000. Le et al (33) and Rani et al (34) solved the problem. in 3D, for Reynolds up to 5.100 and 2.000 respectively. Literature is prolific on both experimental and numerical solutions of the backward-facing step. All of these reaserchers have shown that this flow has very complex characteristics, with layer separation, reattachment and recirculation which occur in many practical engineering situations.

A well known result among all this researchers is that the backward-facing step flow remains laminar until Reynolds 1.200, transitional between 1.200 and 6.600 approximately, and beyond that number it becomes turbulent.

The domain is presented in figure 24 and the 16.000 quadratic triangular elements mesh in 25. These dimensions were selected in order that inlet and outlet boundary conditions do not affect the solution for the Reynolds number taken (31). The mesh is clustered near the walls for better resolving the boundary layer. The inflow condition is the specified (Dirichlet) velocity with a parabolic profile with a maximum value of 1.5, as it is considered in the references. The outflow condition is of pressure and the vertical component of velocity specified, and equal to zero and the walls have non-slip boundary conditions. Reynolds numbers were set to 800, 1.200, 1.600 and 3.000. The timestep selected to iterate is 0.1 for both methods ( $CFL \simeq 2.5$ ).



Figure 24 – Backward facing step problem statement

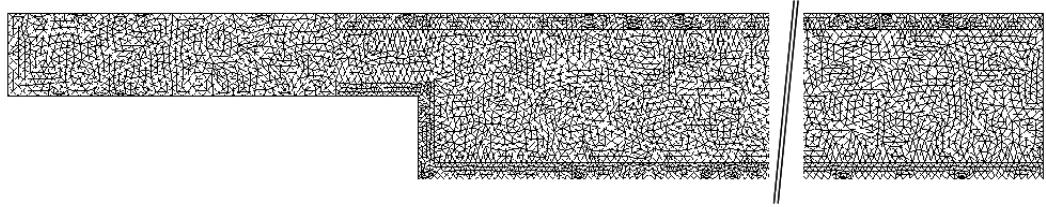


Figure 25 – Backward facing step mesh

#### 4.4.4 Backward facing step test results

This test has shown interesting results about weaknesses of the semi-Lagrangian method, corroborating the ones obtained with the lid-driven cavity test. Tests for low Reynolds numbers (laminar flow) have shown good agreement between both implementations, seen in 26. These results also match the ones found in (30) and (31). As said in the chapter before,  $\Delta t = 0.1$ , giving a  $CFL = 2.5$ .

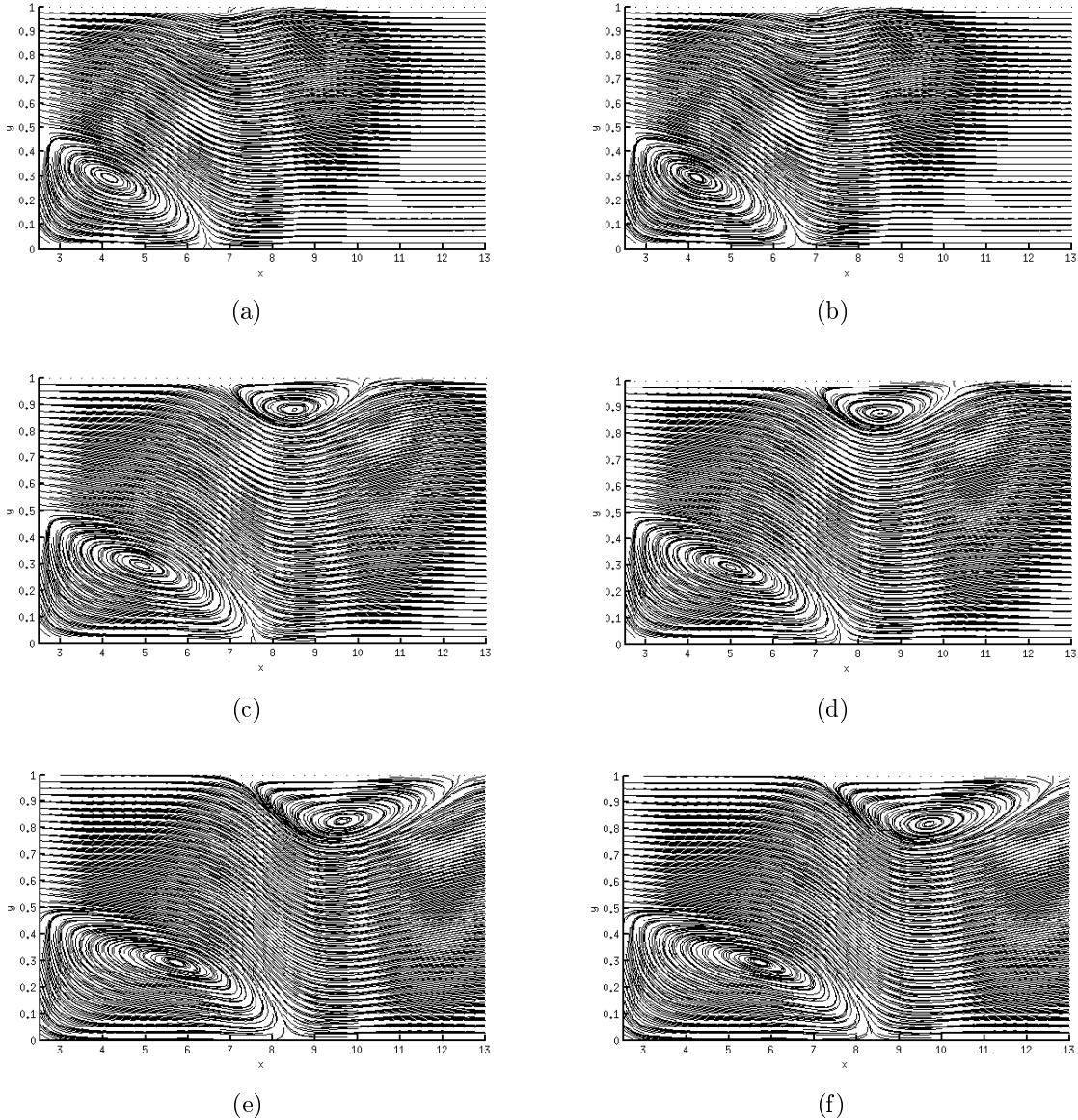


Figure 26 – Semi-Lagrangian (left column) and SUPG results (right columns) for  $Re=800$ ,  $Re=1200$  and  $Re=1600$  (top, middle and bottom line respectively).

For Reynolds = 3.000 (transition region) we have two very thin and very long vortices after the step, as can be seen in (30), (31) and the experimental results of (32). The SUPG gave results that fits with the ones found in literature (see figure 28). The semi-Lagrangian shows numerical instability in the transitional (and turbulent) region. Tests were ran with a very low CFL ( $CFL \simeq 0.01$ , in order to diminish the error of the semi-Lagrangian linear interpolation of the material derivative), offering important data for the analysis. Until  $t = 70$  both methods (SUPG and semi-Lagrangian) gave the same results, but from that point on the semi-Lagrangian begins to show instability in the pressure, which grows with time. Diminishing the  $\Delta t$  (and thus the CFL number)

worsens the problem instead of improving the results. That is why we identify this as a numerical instability of the method. This problem is not solved even with very fine meshes, tests were ran with a 120.000 elements mesh with the same results.

Other test performed to confirm these results was to take the results obtained with the SUPG until some advanced point in the simulation ( $t = 900$ , very close to steady state), and continue with the semi-Lagrangian, to see if it converges to the solution computed by the SUPG (the one seen in (30) and (31)). As the semi-Lagrangian algorithm begins to compute the solution, the instability appears, ending in a result similar as the one seen in 4.27(b).

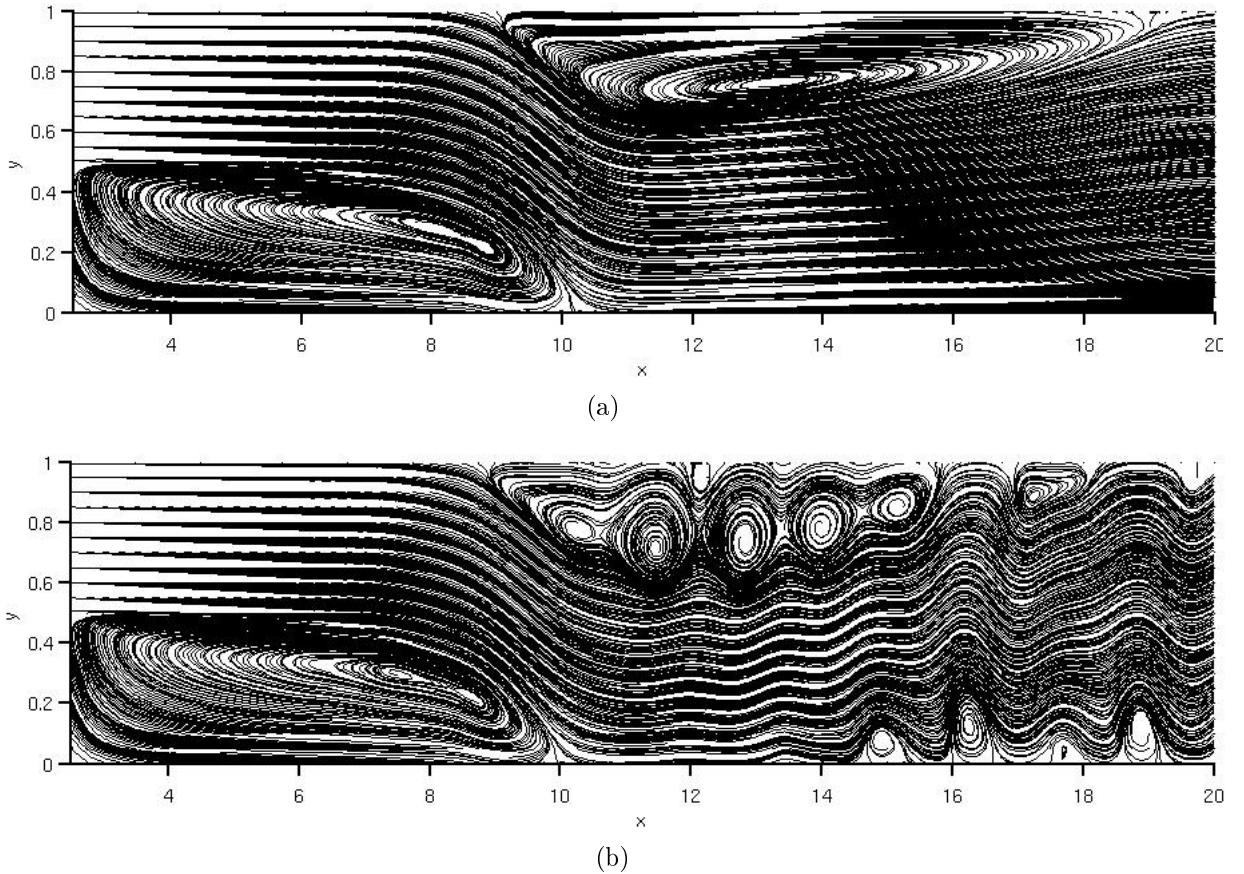


Figure 27 – (a) SUPG and (b) semi-Lagrangian results for an 16.000 elements mesh and  $\Delta t = 0.1$

Results with linear interpolation for velocity in the semi-Lagrangian (section 3.1 ) renders very diffusive results, as those seen in section 4.3.4 . Excessive mesh refinement must be done to counter the numerical diffusion introduced by the linear interpolation. In figure 4.28(b) is shown a result with a 120.000 elements mesh (6 node triangles) still showing very diffusive results.

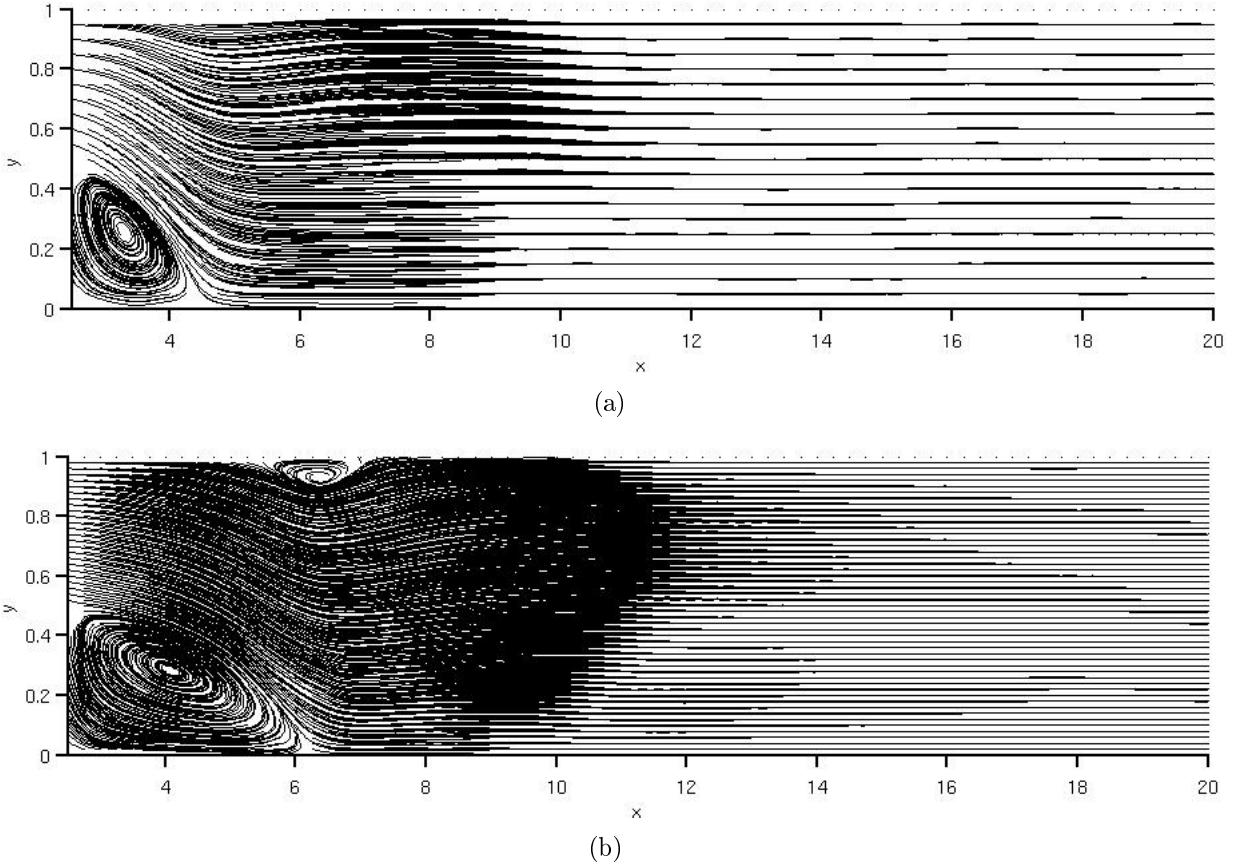


Figure 28 – semi-Lagrangian with linear interpolation ( $Re = 3.000$ ): (a) 16.000 elements mesh; (b) 120.000 elements mesh.

#### 4.4.5 Execution times

Optimizing execution times was not in the scope of this work, however, it is interesting to see the advantages of the parallel implementation and one non-negligible benefit of the semi-Lagrangian method: it is a very fast solver. To obtain results seen in table 3 we have used 50 iterations of the lid-driven cavity test for Reynolds=1.000 and with the 40.000 elements mesh. Parallel simulations were ran with 8 processes, as stated in section 4.2 . It is also interesting to see that the semi-Lagrangian calculation (finding the velocity in the previous timestep of a particle that now occupies one node, for every node in the mesh) is quite expensive: about 44.15% of total runtime, being almost all of the remaining time (55.85%) the linear system solving.

Table 3 – Compared execution times

Method	Serial implementation	Parallel implementation
SUPG	100%	45.84%
Semi-Lagrangian	15.84%	5.76%

## SUMMARY

In this work we have investigated the performance of the semi-Lagrangian method for advection-dominated fluids. We have compared the performance against a very well known method, the Streamline Upwind Petrov Galerkin (SUPG). Both were implemented using the *libMesh* finite element library.

In order to determine the semi-Lagrangian potentiality some benchmark tests were ran on the convection-diffusion and the incompressible Navier-Stokes equations. It demonstrated to be a very fast method (because it linearizes the problem, thus leaving just a linear system to be solved). At low Reynolds numbers it has shown good results for CFLs up to 4. The method presents serious problems for higher Reynolds numbers; it shows oscillations and with very small CFLs it is unstable. Another conclusion of interest is that high order elements must be used to interpolate the particle trajectory, low order elements produce overdiffusive results.

### 5.1 Future Research Area

To improve the previously presented results, the following points deserve further attention:

- Implementation of an higher order semi-Lagrangian: improving the order of the material derivative approximation (particle trajectories).
- Research on the causes of the overdiffusive results when using linear interpolation on the calculation of the properties of the departure point (velocity, concentration of a substance) in the semi-Lagrangian computation.
- Research on the causes of the numerical oscillation (and instability) seen in the method at high Reynolds numbers and low CFLs.

## REFERENCES

- 1 HACHEM, E. et al. Stabilized finite element method for incompressible flows with high Reynolds number. *Journal of Computational Physics*, n. 229, p. 8643–8665, 2010.
- 2 HUGHES, T.; BROOKS, A. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Engrg.*, v. 32, p. 199–259, September 1982.
- 3 TEZDUYAR, T. E. Stabilized finite element formulations for incompressible flow computations. *Advances in applied mechanics*, v. 28, 1991.
- 4 DONEA, J.; HUERTA, A. *Finite element methods for fluid problems*. [S.I.]: John Wiley & Sons Ltd, 2003.
- 5 ZIENKIEWICZ, O.; TAYLOR, R. *The finite element method: Volume 1: The basis*. [S.I.]: Butterworth-Heinemann, 2000.
- 6 ANJOS, G. R. dos et al. Numerical modeling of the hydrodynamic field coupled to the transport of chemical species through the finite element. In: *6th International Congress on Industrial and Applied Mathematics (ICIAM 2007)*. Zurique: 6th International Congress on Industrial and Applied Mathematics, 2007.
- 7 KUNDU, P. K.; COHEN, I. M. *Fluid Mechanics*. 2nd. ed. London and San Diego, CA: Academic Press, 2002.
- 8 ZIENKIEWICZ, O.; TAYLOR, R. *The finite element method: Volume 3: Fluid dynamics*. [S.I.]: Butterworth-Heinemann, 2000.
- 9 BLANK, H.; RUDGYARD, M.; WATHEN, A. Stabilized finite element methods for steady incompressible flow. *Comput. Methods Appl. Engrg.*, n. 174, p. 91–105, 1999.
- 10 TEZDUYAR, T. E.; AKIN, J. E. Calculation of the advective limit of the SUPG stabilization parameter for linear and higher-order elements. *Comput. Methods Appl. Engrg.*, n. 193, p. 1909–1922, 2004.
- 11 SHEU, T. W. H.; LIN, R. K. Newton linearization of the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, v. 44, p. 297–312, 2004.

- 12 SHIN, H. H. *A methodology of study of three dimensional stratified turbulent fluid flow for hydroelectric power plant reservoir simulation*. Dissertação (Mestrado) — Universidade do Estado do Rio de Janeiro, Rio de Janeiro, Brasil, 2009.
- 13 KIRK, B. et al. A C++ library for parallel adaptive mesh refinement/coarsening simulations. *Engineering with Computers*, v. 22, n. 3, p. 237–254, 2006.
- 14 BALAY, S. et al. *PETSc Web page*. 2011. [Http://www.mcs.anl.gov/petsc](http://www.mcs.anl.gov/petsc).
- 15 SHEWCHUK, J. R. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In: LIN, M. C.; MANOCHA, D. (Ed.). *Applied Computational Geometry: Towards Geometric Engineering*. [S.I.]: Springer-Verlag, 1996, (Lecture Notes in Computer Science, v. 1148). p. 203–222. From the First ACM Workshop on Applied Computational Geometry.
- 16 KIRK, B. et al. *libMesh-web-page*. 2002. [Http://libmesh.sourceforge.net/index.php](http://libmesh.sourceforge.net/index.php).
- 17 MPI Web page. 2011. [Http://www.mcs.anl.gov/research/projects/mpi/](http://www.mcs.anl.gov/research/projects/mpi/).
- 18 SLEPc Web page. 2011. [Http://www.grycap.upv.es/slepc/](http://www.grycap.upv.es/slepc/).
- 19 BLAS Web page. 2011. [Http://www.netlib.org/blas/](http://www.netlib.org/blas/).
- 20 LAPACK Web page. 2011. [Http://www.netlib.org/lapack/](http://www.netlib.org/lapack/).
- 21 LASPack Web page. 2011. [Http://www.netlib.org/utk/papers/iterative-survey/node35.html](http://www.netlib.org/utk/papers/iterative-survey/node35.html).
- 22 SI, H. *TetGen Library*. 2011. [Http://tetgen.berlios.de/](http://tetgen.berlios.de/).
- 23 KARYPIS, G.; KUMAR, V. *METIS unstructured graph partitioning and sparse matrix order*. [S.I.], August 1995.
- 24 SAAD, Y.; SCHULTZ, M. H. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, v. 7, n. 3, p. 856–869, 1986.
- 25 MATOS, L. *Desenvolvimento de Modelos 2D para Simulação de Escoamentos Ambientais*. Dissertação (Mestrado) — Universidade do Estado do Rio de Janeiro, Rio de Janeiro, Brasil, 2008.
- 26 SHANKAR, P.; DESHPANDE, M. Fluid mechanics in the driven cavity. *Annu. Rev. Fluid Mech.*, v. 32, p. 93–136, 2000.

- 27 ERKTURK, E.; CORKTE, T.; GÖCÖL, C. Numerical solution of 2-d steady incompressible driven cavity flow at high Reynolds numbers. *International Journal for Numerical Methods in Fluids*, v. 48, p. 747–774, 2005.
- 28 GHIA, U.; GHIA, K.; SHIN, C. High Reynolds solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, v. 48, p. 387–411, 3 1982.
- 29 OBERKAMPF, W. L.; TRUCANO, T. G. Verification and validation in computational fluid dynamics. *Progress in Aerospace Science*, n. 38, p. 209–272, 2002.
- 30 CRUCHAGA, M. A. A study of the backward-facing step problem using a generalized streamline formulation. *Commun. Numer. Engng*, v. 14, p. 697–708, 1998.
- 31 ERKTURK, E. Numerical solutions of 2-d steady incompressible flow over a backward-facing step, part I: High Reynolds numbers solutions. *Computers & Fluids*, v. 37, p. 633–655, 2008.
- 32 LEE, T.; MATEESCU, D. Experimental and numerical investigation of 2-d backward-facing step flow. *Journal of Fluids and Structures*, v. 12, p. 703–716, 1998.
- 33 LE, H.; MOIN, P.; KIM, J. Direct numerical simulation of turbulent flow over a backward-facing step. *J. Fluid Mech.*, v. 330, p. 349–374, 1997.
- 34 RANI, H. P.; SHEU, T. W. H.; TSAI, E. S. F. Eddy structures in a transitional backward-facing step flow. *J. Fluid Mech.*, v. 588, p. 43–58, 2007.
- 35 BALAY, S. et al. *PETSc Users Manual*. [S.I.], 2010.
- 36 DEMBO, R. S.; EISENSTAT, S. C.; STEIHAUG, T. Inexact Newton Methods. *SIAM Journal on Numerical Analysis*, v. 19, n. 2, p. 400–408, April 1982.
- 37 ELIAS, R.; COUTINHO, A.; MARTINS, M. Inexact Newton-type methods for non-linear problems arising from the SUPG/PSPG solution of steady incompressible Navier-Stokes equations. *J. of the Braz. Soc. of Mech. Sci. & Eng.*, XXVI, n. 3, 2004.
- 38 HUGHES, T. *The finite element method: Linear static and dynamic finite element analysis*. New York: Dover Publications, 2000.
- 39 KOZEL, K.; LOUDA, P.; PRIHODA, J. Numerical methods of high order accuracy for incompressible flows. *Mathematics and Computers in Simulation*, v. 80, p. 1734–1745, 2010.

- 40 PIRONNEAU, O. *Finite element methods for fluids*. Paris: John Wiley & Sons and Masson, 1989.
- 41 TEZDUYAR, T. E.; SATHE, S. Stabilization parameters in SUPG and PSPG formulations. *Journal of Computational and Applied Mechanics*, v. 4, n. 1, p. 71–88, 2003.
- 42 TEZDUYAR, T. E.; OSAWA, Y. Finite element stabilization parameters computed from element matrices and vectors. *Comput. Methods Appl. Engrg.*, n. 190, p. 411–430, 2000.
- 43 WRIGHT, N. Introduction to numerical methods for fluid flow. In: BATES, P. D.; N.LANE, S.; FERGUSON, R. I. (Ed.). *Computational Fluid Dynamics: Applications in Environmental Hydraulics*. England: John Wiley & Sons Ltd, 2005. cap. 7, p. 147–68.

## APPENDIX A – Libmesh implementation

We append in a CD-ROM our *libMesh* implementation of the SUPG and the semi-Lagrangian methods to solve the incompressible Navier-Stokes and the scalar transport equations. The code was compiled with MPICH2 version 1.3.1, PETSc version 3.1.6 and *libMesh* revision 4151. C++ compiler is gcc-4.5.

Along with our code, we include the *libMesh*, PETSc and MPICH2 versions used to compile it.