Fourth year internship report
presented by

Eowyn Renoud Hallereau

Major Applied Mathematics and Modelling
Year 2024 - 2025

Numerical simulation of the Navier-Stokes equations using finite element methods

Faculty of Mathematics and Informatics,
Hanoi University of Science and Technology

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF MATHEMATICS AND INFORMATICS

Training supervisor : TA Thi Thanh Mai
Academic advisor : TROMEUR-DERVOUT Damien

January 2025

POLYTECH Lyon - UNIVERSITE CLAUDE BERNARD LYON 1
Domaine Scientifique de La Doua – 15, Boulevard Latarjet
69622 VILLEURBANNE CEDEX
Tél. (33) 04.72.43.12.24 - Fax. (33) 04.72.43.12.25
http://polytech.univ-lyon1.fr/

Acknowledgments

I would like to express my sincerest gratitude to Professor TA Thi Thanh Mai, my training supervisor at the Faculty of Mathematics and Informatics. Her guidance, support, and instruction throughout the entirety of my five-month internship have been of the highest importance for me and for the success of this internship.

Furthermore, I would like to express my appreciation to the Hanoi University of Science and Technology for their warm welcome.

I am particularly thankful to Ms. TRAN Vu Huong Tra and Ms. NGUYEN Phuong Daisy for their assistance with the visa procedures and my arrival in Hanoi.

I would like to extend my gratitude to Professor Tromeur-Dervout Damien for being my academic advisor, and for his assistance in the preparation of this report.

Finally, I would like to acknowledge the administration of Polytech Lyon for their support in navigating the administrative procedures, including the agreement and the acquisition of the regional grant.

Contents

List of Abbreviations and Notations

HUST	Hanoi University of Science and Technology
BNB theorem	Banach-Necas-Babuska theorem
$(x, y)^T$	Transpose of the vector (x, y)
$\nabla u = (\frac{\partial u_1}{\partial x}, \frac{\partial u_2}{\partial y})^T$	Gradient of u
$\nabla . u = \frac{\partial u_1}{\partial x} + \frac{\partial u_2}{\partial y}$	Divergence of u
$\Delta u = \nabla . (\nabla u) = \frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_2}{\partial y^2}$	Laplacian of u
$u \otimes v = (u_1 v_1, u_1 v_2, u_2 v_1, u_2 v_2)^T$	Tensor product
$u_{\|E}$	Restriction of the function u on the set E
$Ker(E)$	Kernel (nullspace) of the set E
$n(x)$	Outward normal
$\mathbb{P}_k^d$	Vector space of polynomials of d variables and of total degree at most k
$\mathcal{L}(E; F)$	Vector space of continuous linear operators from V to W
$L^p(\Omega, \mathbb{K})$	$= \{u : \Omega \to \mathbb{K};\ \int_\Omega \|u(x)\|^p \, dx < +\infty\}$
$W^{s,p}(\Omega)$	$= \{u \in \mathcal{D}'(\Omega);\ \partial^\alpha u \in L^p(\Omega),\ \|\alpha\| \leq s\}$
$H^1(\Omega)$	$= \{u \in L^2(\Omega);\ \partial_i u \in L^2(\Omega), 1 \leq i \leq d\}$
$H_0^1(\Omega)$	$= \{u \in H^1(\Omega);\ u(a) = u(b) = 0\}$
$H^{-1}(\Omega)$	$= (H_0^1(\Omega))'$
$\mathcal{C}^0(\Omega)$	Space of continuous functions on $\Omega \subset \mathbb{R}^2$
$\mathcal{D}(\Omega)$	Space of infinitely differentiable functions compactly supported in Ω
$\|u\|_V$	Norm of u in the normed space V
$\|u\|_{W^{s,p}(\Omega)}$	Norm of u in $W^{s,p}(\Omega)$
$\|u\|_{s,\Omega}$	Norm of u in $W^{s,2}(\Omega) = H^s(\Omega)$
$\|u\|_{1,\Omega}$	Norm of u in $H^1(\Omega)$
$\|u\|_{-1,\Omega}$	Norm of u in $H^{-1}(\Omega)$
$\|u\|_{0,\Omega} = \|u\|_{L^2(\Omega)}$	Norm of u in $L^2(\Omega)$

List of Figures

Introduction

The objective of this internship is to simulate the Navier-Stokes equations using finite element methods and the FreeFem++ software. In particular, the simulations will model a fluid flow around a cylinder and incorporate pollution or a scalar field driven by the advection equation.

The Navier-Stokes problem describes the movement of a Newtonian fluid. It has a number of applications in engineering, physics, medicine, and other fields. The problem also presents a mathematical challenge due to the presence of non-linear terms and the fact that its existence and smoothness have not yet been proven in three dimensions.

A simulation of a flow around an obstacle, such as a cylinder, has a number of potential applications. The simulation can be used to analyse the aerodynamic behaviour of bridge pylons, model the impact of natural obstacles on water flow in rivers, optimise the aerodynamic properties of sports equipment, and study the dynamics of flow around underwater support structures. For instance, offshore monopile wind turbines are supported by a large pylon that penetrates the seabed. In order to construct these structures, it is critical to comprehend the manner in which the monopile foundation interacts with ocean currents, waves and tides. This is in terms of predicting hydrodynamic forces, potential vortex shedding and seabed scouring. Therefore, the results of two-dimensional simulations can inform decisions regarding the orientation of turbines, the distance between structures, and the measures needed to prevent seabed erosion. Furthermore, a simulation of a flow around a cylinder can be employed to ensure the safety of such a construction.

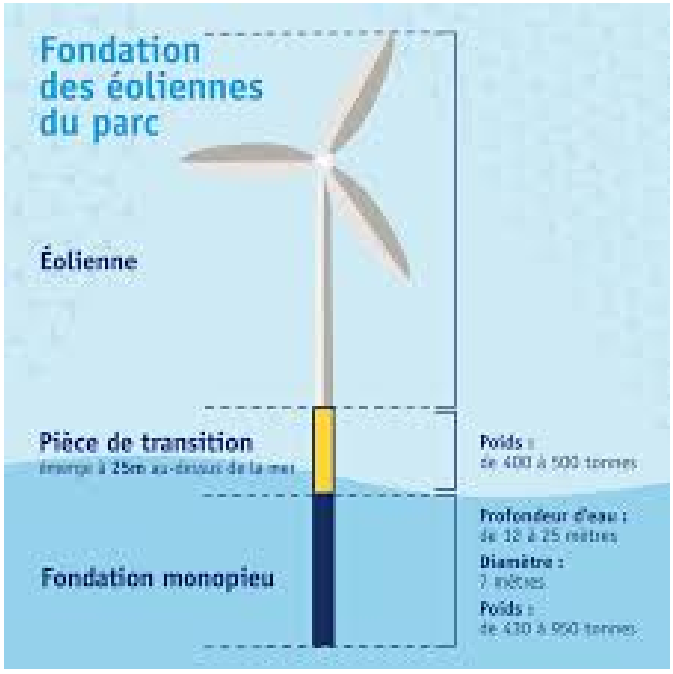


Figure 1: Foundation of a monopile wind turbine [1]

In order to achieve the objective of the internship, several steps have been accomplished. In the first part of the internship, the mathematical models to be simulated were studied, with a primary focus on the Navier-Stokes problem as it is the main subject under consideration. Moreover, the Stokes equations have been examined as they constitute a component of the Navier-Stokes equations. The well-posedness of the problem has been demonstrated. Thereafter, the finite element methods necessary to obtain simulations have been studied from a mathematical perspective, including the characteristics and Galerkin methods. Then, the focus has been on the advection equation and the discontinuous Galerkin method, a more complex finite element method.

After this theoretical part, the simulations were conducted, and the resulting outputs will be presented. The initial simulations included two simulations of the Stokes problem : the driven cavity test and a simulation of an exact solution. Subsequently, simulations of the Navier-Stokes problem have been explored. In order to validate the programs, well-known simulations have been conducted, such as the cavity test and the Taylor vortex problem. Prior to the presentation of more complex simulations, a preliminary simulation involving only the advection equation was conducted. The purpose was to validate the program that uses the discontinuous Galerkin method. Finally, simulations of flow around obstacles were conducted, and the results for simulations using circular or square cylinders as obstacles will be presented. A simulation combining the Navier-Stokes equations and the advection equation has been conducted, representing the most complex simulation.

Chapter 1

Work environment

In the context of my studies at Polytech Lyon engineering school, I completed a work placement in association with my major in Applied Mathematics and Modelling. In selecting this internship in a research laboratory, I was motivated by my desire to orient my studies towards research. I am considering undertaking a PhD after my engineering diploma. I also decided to conduct my internship in an international location, specifically in Vietnam, since I had a strong desire to experience a new culture and to live in a country significantly different from France. This decision also allowed me to validate my international mobility requirement for my diploma. The entire internship was conducted in English.

1.1 Host institution and research group

My work placement was conducted at the Hanoi University of Science and Technology (HUST), specifically within the Faculty of Mathematics and Informatics and the research group Optimisation and Scientific Computing, under the supervision of Professor TA Thi Thanh Mai.

The Hanoi University of Science and Technology is a prominent university in Vietnam. Established in 1956, HUST comprises a total of 11 schools and faculties, in addition to 6 research centres.

The Faculty of Mathematics and Informatics is composed of 6 departments, also referred to as expert groups, each of which is responsible for teaching and research. The 6 departments are :

- Data Science and Application
- Optimisation and Scientific Computing
- Mathematical Foundations for Informatics and Information Systems
- Probability and Statistics and Applications
- Calculus
- Algebra

Figure 1.1 presents an organisation chart of the Faculty.

The department of Optimisation and Scientific Computing is an expert group within the faculty. My tutor is responsible for the department, and it comprises 16 members, including professors, associate professors and administrative staff. The department is responsible for teaching a number of courses, including the basic modules of the Applied Mathematics orientation, and courses in the undergraduate training programme of the Faculty of Mathematics.

However, the department also functions as a research group. The principal research topics of the department are as follows:

- Dynamic systems and their applications
- Theory of partial differential equations
- Numerical methods for partial differential equations
- Mathematical models applied in complex systems, such as biomathematics
- Mathematical models in artificial intelligence and deep learning

The researchers meet approximately once a month for a seminar, with a different topic being discussed at each session. This ensures that all researchers are informed of the latest research and the subjects of their colleagues. In the context of their research activities, the team has established cooperative relationships with several universities, including those in Hanoi, as well as some in other cities in Vietnam, and international institutions. For instance, they have collaborative links with French universities such as Sorbonne University, University of Lorraine, and University of Bordeaux.

1.2 Working conditions

My internship was characterised by a high degree of autonomy. I had the flexibility to work remotely from home, at the university library on campus, or at other locations such as cafes, as the research group does not have a large office with desks for everyone. Furthermore, I was free to determine my own working hours and organise my schedule according to my preferences. However, I didn't have much interaction with other members of the research group, which is a regrettable point.

I attended a single meeting per week with Ms. Mai, for a duration of 30 minutes to 1 hour. The weekly appointment commenced with a presentation of my work of the week, in which I explained my successful accomplishment and understanding, and what had been more complex. Thereafter, Ms. Mai gave me new directives for the following week, in order to help me progress one step at a time on my subject. Due to the considerable distance between Ms. Mai's home and the university, and the really heavy traffic in Hanoï, the weekly meetings were held via Teams starting in December.

During the work placement, I attended a seminar organised by the department because the subject, which was the discontinous Galerkin method, was relevant to my own studies. However, as the seminar was conducted largely in Vietnamese, I was unable to understand everything. In addition, I also prepared a presentation of 15 minutes on a component of my subject, the Navier-Stokes equations, to discuss and exchange with other students of Ms. Mai. However, due to scheduling conflicts, I was unfortunately unable to share this presentation. At the end of the placement, I wrote a short article for a journal, on my final simulation, which combined Navier-Stokes equations and the advection equation. However, this article remains in progress.

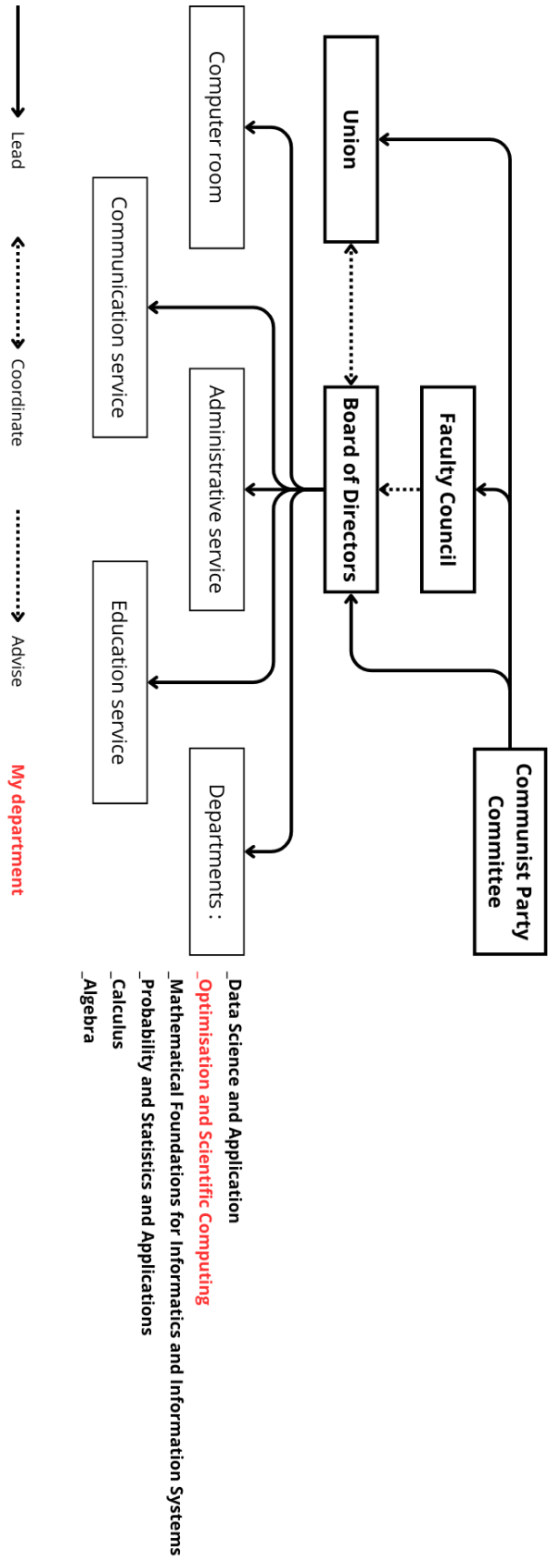


Figure 1.1: Organisation chart of the Faculty of Mathematics and Informatics

Chapter 2

Mathematical models

The focus of this chapter is the mathematical models that will be used in the simulations. The subject is the simulations of the Navier-Stokes equations, therefore, detailed information regarding these equations will first be provided. As numerical simulations are to be conducted, it is necessary to expose numerical methods to deal with the complex equations depending on time and space. Then, the advection equation will be presented, as it is also an important equation that will be used in the simulations.

2.1 The Navier-Stokes problem

The Navier-Stokes problem is a set of equations used to describe fluid flows. In some cases, it can be simplified to form the Stokes problem. Finite element methods are employed to obtain numerical simulations of these problems. For this section, more detailed regarding the Navier-Stokes equations can be found in [22], [13] and [14]. For a more detailed exposition of the Stokes problem, readers are directed to [5]. For a more comprehensive overview on the various finite element methods, see references [10] and [11]. For precise information on the characteristics method, see [2] and [20].

2.1.1 The origin of the equations

The Navier-Stokes equations are constitute of two fundamental equations : the mass conservation equation and the momentum conservation equation. The following section will explain the origin of these equations. For that, let $u(t,x)$ be the velocity of the fluid and $\rho(t,x)$ its density.

The principle of mass conservation states that the variation in mass within a fluid must be equal to the input mass flux at the boundaries of the fluid. This can be expressed in equation form as follows :

$$\begin{aligned} \frac{d}{dt}\left(\int_\Omega \rho\, dx\right) &= -\int_{\partial\Omega} \rho u.n\, ds \\ \Leftrightarrow \frac{d}{dt}\left(\int_\Omega \rho\, dx\right) &= -\int_\Omega \nabla.(\rho u)\, dx \quad \text{(Green-Ostrogradski, see 2.2)} \\ \Leftrightarrow \frac{\partial \rho}{\partial t} + \nabla.(\rho u) &= 0 \quad \text{in} \quad]0,+\infty[\times\Omega. \end{aligned} \tag{2.1}$$

The Green-Ostrogradski formula can be written as follows :

$$\int_{\partial\Omega} v.n\, ds = \int_\Omega \nabla.v\, dx, \quad \forall \text{ function } v. \tag{2.2}$$

The fundamental principle of momentum conservation asserts that the variations in momentum must be equal to the forces acting upon the fluid. This signifies that:

$$\frac{d}{dt}\left(\int_\Omega \rho u\, dx\right) = F_{ext} + F_{int}$$

$$\Leftrightarrow \int_\Omega \left(\frac{\partial(\rho u)}{\partial t} + \nabla(\rho u \otimes u)\right) dx = F_{ext} + F_{int} \quad \text{(Reynolds transport formula, see 2.4).}$$

But $\frac{\partial(\rho u)}{\partial t} + \nabla.(\rho u \otimes u) = \frac{\partial \rho}{\partial t}u + \rho\frac{\partial u}{\partial t} + \nabla.(\rho u)u + \rho(u.\nabla)u = \rho\frac{\partial u}{\partial t} + \rho(u.\nabla)u$ because of (2.1). So for the principle of momentum conservation, the equation is :

$$\rho\left(\frac{\partial u}{\partial t} + (u.\nabla)u\right) = F_{ext} + F_{int} \quad \text{in} \quad]0, +\infty[\times\Omega. \tag{2.3}$$

The forces applied are the external forces (such as gravity) denoted f, and the internal forces: $\nu\Delta u - \nabla p$, where ν is the kinematic viscosity, and p the pressure. The Reynolds transport formula can be expressed as follows : let $q(t, x)$ be any function :

$$\frac{d}{dt}\left(\int_{\Omega(t)} q\, dx\right) = \int_{\Omega(t)} \left(\frac{\partial q}{\partial t} + \nabla.(qu)\right) dx \quad \forall q, \tag{2.4}$$

where $u(t, x)$ is the velocity.

In this work, we will consider incompressible fluids, which means that the density ρ is constant, and Newtonian fluids, which are characterised by a constant viscosity ν regardless of stress or rate. Using the assumptions just mentioned, the incompressible Navier-Stokes problem can be derived from equations (2.1) and (2.3) as follows :

$$\begin{cases} \dfrac{\partial u}{\partial t} + u.\nabla u - \nu\Delta u + \nabla p = f, \\ \nabla.u = 0. \end{cases} \tag{2.5}$$

2.1.2 The Stokes problem

When the steady Navier-Stokes equations are linearised, the Stokes equations are obtained. These equations govern the flow of fluids in steady state, characterised by velocities that are very slow and Reynolds numbers that are significantly less than one. In such cases, the inertial forces are assumed to be negligible in front of viscous forces. The Stokes problem can be written in the form : find $u \in V$ and $p \in Q$ such that

$$\begin{cases} -\Delta u + \nabla p = f, \\ \nabla.u = 0, \end{cases} \tag{2.6}$$

where u is the velocity of the fluid, p is its pressure, and f is a body force acting on the fluid. In order to define the boundary conditions, we shall add a homogeneous Dirichlet condition : $u = 0$ on $\partial\Omega$.

The variational problem

The problem (2.6) can be expressed in the so-called weak form by multiplying the equations by a test function v and integrating over the domain Ω. The problem is then : find $u \in V$ and $p \in Q$ such that

$$\begin{cases} \int_\Omega \nabla u.\nabla v\, dx - \int_\Omega p\nabla.v\, dx = \int_\Omega fv\, dx, \quad \forall v \in V, \\ -\int_\Omega q\nabla.u\, dx = 0, \quad \forall q \in Q. \end{cases} \tag{2.7}$$

For convenience, the problem (2.7) is rewritten using $V = [H_0^1(\Omega)]^2$, $Q = L^2(\Omega)$ and the bilinear forms $a : V \times V \to \mathbb{R}$ and $b : V \times Q \to \mathbb{R}$:

$$a(u, v) = \int_\Omega \nabla u.\nabla v\, dx, \tag{2.8}$$

$$b(v, q) = -\int_\Omega q\nabla.v\, dx. \tag{2.9}$$

The problem (2.7) is then equal to : find $u \in V$ and $p \in Q$ such that

$$\begin{cases} a(u, v) + b(v, p) = (f, v), \quad \forall v \in V, \\ b(u, q) = 0, \quad \forall q \in Q, \end{cases} \tag{2.10}$$

where $(f, v) = \int_\Omega fv\, dx$.

The spaces $V = [H_0^1(\Omega)]^2$ and $Q = L^2(\Omega)$ are reflexive Banach spaces and $f \in V' = \mathcal{L}(V; \mathbb{R})$ (see Appendix A.1 for definitions).

The well-posedness

Definition 2.1.1. *The problem (2.10) is said to be well-posed if it admits a unique solution and* $\exists c_1, c_2$ *such that,* $\forall f \in [H^{-1}(\Omega)]^2$ *and* $g \in L^2(\Omega)$*,*

$$\|u\|_{1,\Omega} + \|p\|_{0,\Omega} \leq c_1\|f\|_{-1,\Omega} + c_2\|g\|_{0,\Omega}. \tag{2.11}$$

It is important to demonstrate that the problem (2.10) is well-posed. In order to achieve this, the problem will be rewritten in accordance with the requirements of the saddle-point theory. Two operators, A and B, are defined as follows : $A : V \to V'$ with $\langle Au, v\rangle_{V',V} = a(u, v)$ and $B : V \to Q'$ with $\langle Bv, q\rangle_{Q',Q} = b(v, q)$. Let us also define the nullspace of B : $Ker(B) = \{v \in V;\ \forall q \in Q,\ b(v, q) = 0\}$. Consequently, the problem (2.10) can be stated as follows : find $u \in V$ and $p \in Q$ such that

$$\begin{cases} Au + B^T p = f, \\ Bu = 0. \end{cases} \tag{2.12}$$

In order to prove the well-posedness, an adaptation of the BNB theorem (see Theorem 12) for the saddle-point problem is stated.

Theorem 2.1.2. *Problem (2.10) is well-posed if and only if*

$$\begin{cases} \exists \alpha > 0,\ \inf\limits_{u \in Ker(B)} \sup\limits_{v \in Ker(B)} \dfrac{a(u, v)}{\|u\|_V \|v\|_V} \geq \alpha, \\ \forall v \in Ker(B),\ (\forall u \in Ker(B),\ a(u, v) = 0) \Rightarrow (v = 0), \end{cases} \tag{2.13}$$

and

$$\exists \beta > 0,\ \inf_{q \in Q} \sup_{v \in V} \frac{b(v, q)}{\|v\|_V \|q\|_Q} \geq \beta. \tag{2.14}$$

Proof. To prove the well-posedness of the Stokes equations, Theorem 2.1.2 is applied with $B = \nabla. : V \to Q$ and $Ker(B) = V_0 = \{v \in V;\ \nabla.v = 0\}$.

The bilinear form a is coercive on V because of Poincaré's inequality : if Ω is bounded and open then $\exists c > 0$ such that : $\forall v \in V,\ c\|v\|_V \leq \|\nabla v\|_V$. If the bilinear form a is coercive on V then it is coercive on V_0, which implies condition (2.13).

The condition (2.14) is equivalent to

$$\inf_{q \in Q} \sup_{v \in V} \frac{\int_\Omega q \nabla.v \, dx}{\|v\|_{1,\Omega} \|q\|_{0,\Omega}} \geq \beta, \tag{2.15}$$

and because Q is reflexive and B is surjective, $\beta > 0$ exists. □

2.1.3 The numerical methods

In order to achieve a numerical simulation of the Navier-Stokes equations, several steps are required. The initial step is to discretise by time using the method of characteristics. The second step is to discretise by space the Stokes-like problem obtained at each time step. For this, a number of finite element methods can be used, one of which is the Galerkin method.

The method of characteristics

The first step is to discretise the time interval, with T being the final time and N the number of iterations desired, it becomes : $[0, T] = \bigcup_{n=0}^{N} [t_n, t_{n+1}]$, where $t_n = n\Delta t$ and $\Delta t = \frac{T}{N}$.

In order to utilise the method of characteristics, the Navier-Stokes problem (2.5) is rewritten in terms of the Lagrangian or material derivative :

$$\begin{cases} \dfrac{Du}{Dt} - \nu \Delta u + \nabla p = f, \\ \nabla.u = 0, \end{cases} \tag{2.16}$$

where the material derivative is : $\dfrac{Du}{Dt} = \dfrac{\partial u}{\partial t} + u.\nabla u$.

The Navier-Stokes equations (2.16) are now integrated for each small time interval, along the trajectories of the fluid. In this context, the notation $X(x, t_{n+1}; t)$ represents the position of a fluid particle at time t that reaches the point x at time t_{n+1}. This is the point at time t corresponding to the characteristic curve of the material derivative (see Figure 2.1). The trajectory of this particle is governed by the equations :

$$\begin{cases} \dfrac{dX}{dt}(x, t_{n+1}; t) = u(X(x, t_{n+1}; t), t), \\ X(x, t_{n+1}; t_{n+1}) = x. \end{cases} \tag{2.17}$$

Using these notations, the material derivative can be approximated as follows :

$$\frac{Du}{Dt}(x, t_{n+1}) \approx \frac{u(x, t_{n+1}) - u(X(x, t_{n+1}; t_n), t_n)}{\Delta t}, \tag{2.18}$$

where $X(x, t_{n+1}; t_n) = x - \Delta t.u(x, t_n)$.

Then using an Euler implicit scheme, the equations (2.16) can be rewritten as :

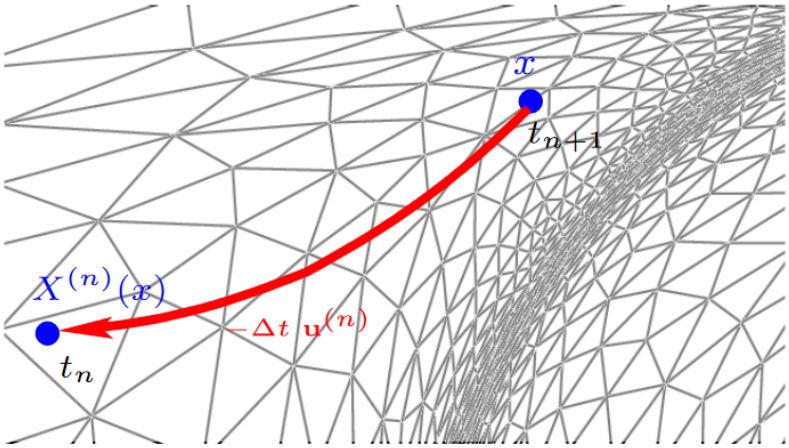


Figure 2.1: The characteristic curve [22]

$$\begin{cases} \dfrac{u^{n+1} - u^n(X^n)}{\Delta t} - \nu \Delta u^{n+1} + \nabla p^{n+1} = f^{n+1}, \\ \nabla . u^{n+1} = 0. \end{cases} \tag{2.19}$$

By the method of characteristics, at each time step, the problem is now a Stokes-like problem, which can be resolved using a finite element method :

$$\begin{cases} u - \nu \Delta u + \nabla p = f, \\ \nabla . u = 0. \end{cases} \tag{2.20}$$

The problem (2.20) will be rewritten in its weak form, as seen previously in section 2.1.2, and it has been demonstrated that this problem is well-posed.

The Galerkin method

The second step is to discretise in space the Stokes problem obtained. In this case, the finite element method used is the Galerkin method. This method is based on replacing the infinite-dimensional spaces V and Q in (2.10) by finite-dimensional spaces V_h and Q_h to obtain the discrete problem : find $u_h \in V_h$ and $p_h \in Q_h$ such that

$$\begin{cases} a_h(u_h, v_h) + b_h(v_h, p_h) = (f_h, v_h), \quad \forall v_h \in V_h, \\ b_h(u_h, q_h) = 0, \quad \forall q_h \in Q_h. \end{cases} \tag{2.21}$$

The approximation is said to be conforming if $V_h \subset V$ and $Q_h \subset Q$. It is important to verify that the velocity discrete space V_h and the pressure discrete space Q_h are compatible, in order to ensure that the discrete problem is well-posed.

Proposition 2.1.3. *If the bilinear form a_h is coercive, the discrete problem (2.21) is well-posed if and only if the spaces V_h and Q_h satisfy the compatibility condition*

$$\exists \beta_h > 0, \quad \inf_{q_h \in Q_h} \sup_{v_h \in V_h} \frac{\int_\Omega q_h \nabla . v_h \, dx}{\|q_h\|_{0,\Omega} \|v_h\|_{1,\Omega}} \geq \beta_h. \tag{2.22}$$

In the present case, $a_h(.,.) = a(.,.)$ and $b_h(.,.) = b(.,.)$ as respectively defined in (2.8) and (2.9). As demonstrated in section 2.1.2, the bilinear form a is coercive.

There are various pairs of finite element spaces, however not all of them can satisfy the condition (2.1.3). One of the most elementary pairs is the $\mathbb{P}_1/\mathbb{P}_1$ finite element, where both the velocity and pressure spaces are approximated by $\mathbb{P}_1$ polynomials. Unfortunately, this finite element does not satisfy the condition (2.1.3). The main reason is that the velocity space is not rich enough. To rectify this issue, the use of the $\mathbb{P}_1$-bubble/$\mathbb{P}_1$ finite element is proposed. A degree of freedom is added per element associated with the barycentre of each element within the mesh (for example, each triangle of the mesh). Another finite element that is more frequently used and that satisfies the condition (2.1.3) is the Taylor-Hood finite element ($\mathbb{P}_2/\mathbb{P}_1$). This is, assuming that Ω is a polyhedron, approximating the pressure by $\mathbb{P}_1$ polynomials, and the velocity by means of $\mathbb{P}_2$ polynomials. Then the approximation spaces are defined as :

$$V_h = \{u_h \in [\mathcal{C}^0(\overline{\Omega})]^2;\ \forall K \in \mathcal{T}_h,\ u_h \circ T_K \in [\mathbb{P}_2]^2;\ u_{h|\partial\Omega} = 0\}, \tag{2.23}$$

$$Q_h = \{p_h \in L^2(\Omega) \cap \mathcal{C}^0(\overline{\Omega});\ \forall K \in \mathcal{T}_h,\ p_h \circ T_K \in \mathbb{P}_1\}, \tag{2.24}$$

where $\mathcal{T}_h$ is a mesh $\mathcal{T}$ of Ω with a mesh size h. For $k \in \mathbb{N}, \mathbb{P}_k$ is the space of the polynomials in one variable and of degree at most k.

Definition 2.1.4. *A polyhedron is defined as a domain with a boundary that is a finite union of polygons. A polygon is a two-dimensional domain whose boundary is a finite union of segments. In the majority of cases, the term polyhedron is used to refer to both polygons and polyhedrons.*

In consequence of the preceding steps, a simulation of the Navier-Stokes equations can be performed.

2.2 The advection equation

In this section, the advection equation will be discussed with reference to [20]. For an exhaustive explanation of the discontinuous Galerkin method, readers are referred to [8].

2.2.1 The equation

The unsteady advection equation, also referred to as the transport equation or continuity equation, describes the transport of a quantity, in this case a fluid. The unknown function ϕ is a characteristic of the fluid, and can be identified as the density, a concentration, or the velocity of the fluid. The problem can be written as :

$$\begin{cases} \dfrac{\partial \phi}{\partial t} + u.\nabla\phi = 0 \text{ in } \Omega, \\ \phi = 0 \text{ on } \partial\Omega^-, \end{cases} \tag{2.25}$$

where u is an advective field and $\partial\Omega^-$ corresponds to the inflow part of the boundary of Ω :

$$\partial\Omega^- = \{x \in \partial\Omega \,|\, u(x).n(x) < 0\}. \tag{2.26}$$

2.2.2 The discontinuous Galerkin method

A comparable discrete problem, similar to the continuous Galerkin method, will be used in the numerical simulations. However, for this method, additional definitions regarding the mesh are necessary.

Definition 2.2.1. *Let $\mathcal{T}_h$ a mesh of Ω. A subset F of $\overline{\Omega}$ is a mesh face if it is not empty, and if :*

1. *There are two distinct mesh elements T_1 and T_2 such that $F = \partial T_1 \cap \partial T_2$, then F is an interface. $\mathcal{F}_h^i$ is the set of the interfaces.*
2. *There is a mesh element $T \in \mathcal{T}_h$ such that $F = \partial T \cap \partial\Omega$, then F is a boundary face. $\mathcal{F}_h^b$ is the set of the boundary faces.*

Definition 2.2.2. *Let v be a function defined on Ω.*

1. *The average of v is defined as :*
$$\{\{v\}\}_F(x) = \frac{1}{2}\left(v_{|T_1}(x) + v_{|T_2}(x)\right). \tag{2.27}$$
2. *The jump of v is defined as :*
$$[[v]]_F(x) = v_{|T_1}(x) - v_{|T_2}(x). \tag{2.28}$$

For $k \in \mathbb{N}$, $\mathbb{P}_k^d$ is the space of the polynomials in d variables and of total degree at most k.

Definition 2.2.3. *The broken polynomial space $\mathbb{P}_k^d(\mathcal{T}_h)$ is defined by :*
$$\mathbb{P}_k^d(\mathcal{T}_h) = \{v \in L^2(\Omega) \,|\, \forall T \in \mathcal{T}_h, v_{|T} \in \mathbb{P}_k^d(T)\}. \tag{2.29}$$

For the discontinuous Galerkin method, the function space V of ϕ and v is a broken finite element space, V_h. This necessitates the introduction of additional terms to the weak form in order to ensure consistency and continuity at the mesh interfaces. The discrete problem for the advection equation (2.25) in its weak form is given by : find $\phi_h \in V_h = \mathbb{P}_k^d(\mathcal{T}_h)$ such that
$$a_h(\phi_h, v_h) = 0 \quad \forall v \in V_h, \tag{2.30}$$
with the discrete bilinear form a_h, which includes an implicit Euler scheme, defined by :
$$a_h(\phi_h, v_h) = \int_\Omega \frac{\phi_h^{n+1} - \phi_h^n}{\Delta t} v_h \, dx + \int_\Omega u.\nabla_h \phi_h^{n+1} v_h \, dx - \int_{\mathcal{F}_h^i} \frac{1}{2}(|u.n| - u.n)[[\phi_h^{n+1}]] v_h \, ds. \tag{2.31}$$

2.2.3 The well-posedness

Proposition 2.2.4. *Let $\|.\|$ be an arbitrary norm. The discrete problem (2.30) is well-posed if and only if the discrete inf-sup condition is true, which means : $\exists C > 0$ such that*
$$C \leq \inf_{v_h \in V_h \setminus \{0\}} \sup_{w_h \in V_h \setminus \{0\}} \frac{a_h(v_h, w_h)}{\|v_h\| \|w_h\|}. \tag{2.32}$$

Proof. To demonstrate the well-posedness of the discrete advection equation, Proposition 2.2.4 can be employed. However, a more straightforward approach to proving well-posedness is to prove the coercivity of the discrete bilinear form a_h defined in (2.31). Indeed, the coercivity of a_h is defined as follows : $\exists C > 0$ such that

$$\forall v_h \in V_h, \quad C\|v_h\|^2 \leq a_h(v_h, v_h). \tag{2.33}$$

This implies that $\forall v_h \in V_h \backslash \{0\}$:

$$C\|v_h\| \leq \frac{a_h(v_h, v_h)}{\|v_h\|} \leq \sup_{w_h \in V_h \backslash \{0\}} \frac{a_h(v_h, w_h)}{\|w_h\|}. \tag{2.34}$$

Thus, condition (2.34) is equivalent to the condition (2.32) in Proposition 2.2.4.

Therefore, let us prove the coercivity. We assume that $\exists \mu \in \mathbb{R}, \mu > 0$ such that

$$\Lambda = -\frac{1}{2}\nabla.u \geq \mu \text{ in } \Omega. \tag{2.35}$$

Then, a_h can be rewritten as : $\forall v_h \in V_h$

$$a_h(v_h, v_h) = \int_\Omega \Lambda v_h^2 \, dx + \int_{\partial\Omega} \frac{1}{2}|u.n|v_h^2 \, ds. \tag{2.36}$$

And, by the assumption (2.35), we have the coercivity of a_h on V_h:

$$\forall v_h \in V_h, \quad a_h(v_h, v_h) \geq C\|v_h\|_c, \tag{2.37}$$

with $C = 1$ and the norm $\|.\|_c$ defined by $\|v\|_c^2 = \mu\|v\|_{L^2(\Omega)} + \int_{\partial\Omega} \frac{1}{2}|u.n|v^2 \, ds$. □

Three problems have been identified : the Stokes equations, the Navier-Stokes equations and the advection equation. The well-posedness of these equations has been established, and the finite element methods that will be employed have been presented. Consequently, it is now possible to discuss the conducted simulations and their outcomes.

Chapter 3

Numerical simulations and results

All simulations were conducted using the FreeFem++ software. FreeFem++ is an open-source solver for partial differential equations (PDEs) utilising finite element methods. The language is analogous to C++, but with a number of additional keywords which are used to construct meshes, finite element spaces, and solve the equations. All the FreeFem++ scripts can be found in Appendix B. The documentation of the FreeFem++ software can be found in [16].

3.1 Simulations of Stokes problem

In this section, the Stokes problem will be simulated, as it is the simplest problem to begin with in our subject. Readers may refer to [16] and [26], for more detailed basic scripts.

3.1.1 The driven cavity test case

The initial simulation conducted, is the driven cavity test case based on the Stokes equations. The velocity field u and the pressure p satisfy

$$\begin{cases} -\Delta u + \nabla p = 0 \text{ in } \Omega, \\ \nabla . u = 0 \text{ in } \Omega. \end{cases} \tag{3.1}$$

The cavity is the unit square $\Omega =]0,1[\times]0,1[$ with some Dirichlet boundary conditions as follows : $u = (1,0)$ on the upper boundary (Γ_3), and $u = 0$ on every other boundary ($\Gamma_1, \Gamma_2, \Gamma_4$) (see Figure 3.1). For the Stokes problem, a grid size of 30×30 is enough to obtain results. In all the simulations, the finite element $\mathbb{P}_2/\mathbb{P}_1$ will be used, however, for this simulation, the $\mathbb{P}_1$-bubble/$\mathbb{P}_1$ element is sufficient. In order to draw a comparison between the two, Figure 3.2 presents the result with $\mathbb{P}_1$-bubble/$\mathbb{P}_1$ element, and Figure 3.3 presents the outcome with $\mathbb{P}_2/\mathbb{P}_1$.

As illustrated in Figure 3.3, the outcomes observed align with the anticipated results for this well-established test. These findings are comparable to those documented in [3]. In the future section 3.2.1, the driven cavity test case for the Navier-Stokes equations will be conducted. The positions of the centre of the main vortex will be compared for different meshes for a more precise analysis.

3.1.2 An exact solution

In order to analyse the evolution of the error in the finite element solution when reducing the mesh size (n $\times$ n), a Stokes problem (3.2) is simulated with a right-hand side and boundary solutions chosen

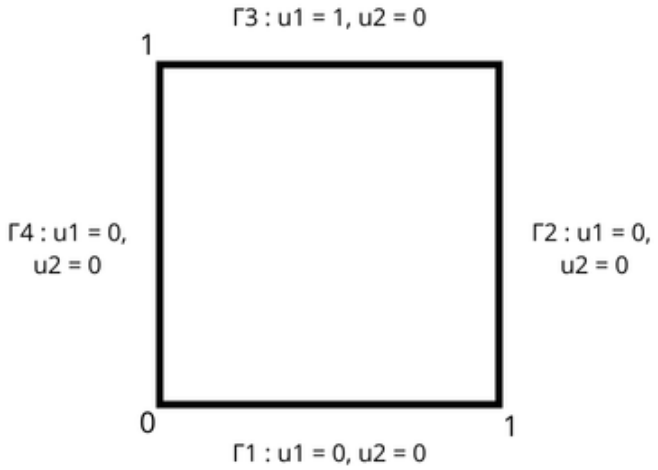


Figure 3.1: Boundaries conditions for the cavity test

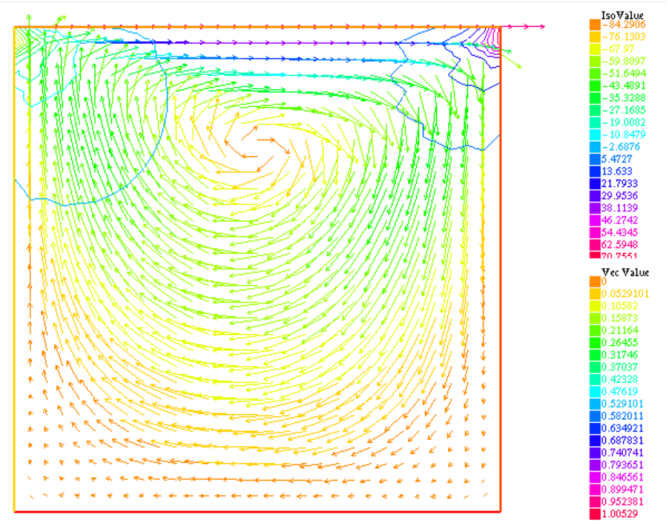


Figure 3.2: Fluid velocity and pressure under Stokes equations with $\mathbb{P}_1 b/\mathbb{P}_1$

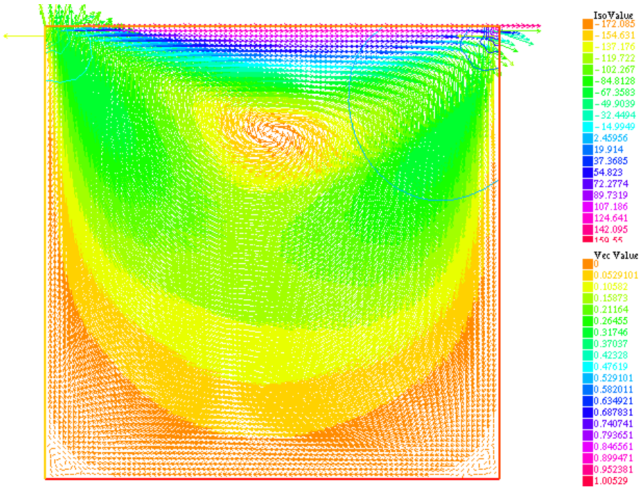


Figure 3.3: Fluid velocity and pressure under Stokes equations with $\mathbb{P}_2/\mathbb{P}_1$

such that an exact solution can be calculated, as in [18]. Then, the exact solution is compared with the computed solution by calculating the following norms :

$$\|u_1 - u_{1h}\|_{L^2(\Omega)}, \quad \|u_2 - u_{2h}\|_{L^2(\Omega)}, \quad \|p - p_h\|_{L^2(\Omega)}.$$

The Stokes problem is : find $u \in V$ and $p \in Q$ such that

$$\begin{cases} -\Delta u + \nabla p = f \text{ in } \Omega, \\ \nabla . u = 0 \text{ in } \Omega, \\ u = g \text{ on } \partial\Omega, \end{cases} \tag{3.2}$$

with $\Omega =]0,1[\times]0,1[$ and $f = (f1, f2)^T$, $g = (g1, g2)^T$ defined by

$$\begin{aligned} f_1(x,y) &= 2\sin(\pi(x+y)) + \frac{1}{\pi}\cos(\pi(x+y)), \\ f_2(x,y) &= -2\sin(\pi(x+y)) + \frac{1}{\pi}\cos(\pi(x+y)), \\ g_1(x,y) &= \frac{1}{\pi^2}\sin(\pi(x+y)), \\ g_2(x,y) &= -\frac{1}{\pi^2}\sin(\pi(x+y)). \end{aligned}$$

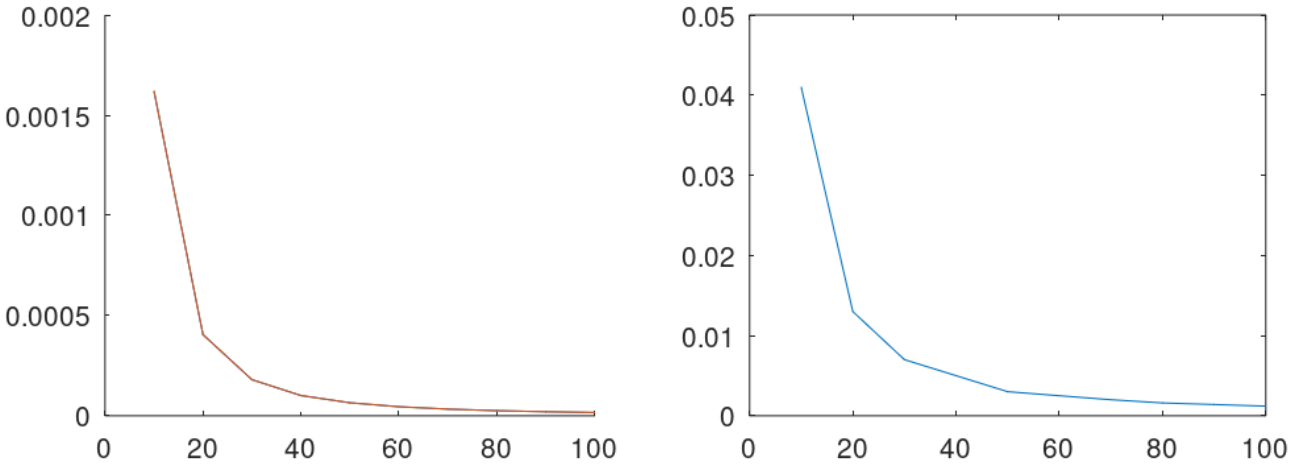
Therefore, the exact solution is as follows :

$$\begin{cases} u_1(x,y) = \dfrac{1}{\pi^2}\sin(\pi(x+y)), \\ u_2(x,y) = \dfrac{1}{\pi^2}\sin(\pi(x+y)), \\ p(x,y) = \dfrac{1}{\pi^2}\sin(\pi(x+y)). \end{cases} \tag{3.3}$$

The velocity obtained with a mesh size of 30×30 and a $\mathbb{P}_1$-bubble/$\mathbb{P}_1$ element is shown in Figure 3.5. Figure 3.4 and Table 3.1 illustrate the convergence of the finite element solution when the mesh is refined, from 10×10 to 100×100. The errors on u_1, u_2, and p, calculated using the L^2 norm, demonstrate a decrease, indicating the efficacy of the program.

Mesh	$\|u - u_h\|_{L^2(\Omega)}$	$\|p - p_h\|_{L^2(\Omega)}$
10×10	1.62e-03	0.041
20×20	4.05e-04	0.013
30×30	1.80e-04	0.007
40×40	1.01e-04	0.005
50×50	6.48e-05	0.003
60×60	4.50e-05	0.0025
70×70	3.31e-05	0.0020
80×80	2.53e-05	0.0016
90×90	2.01e-05	0.0014
100×100	1.62e-05	0.0012

Table 3.1: Error in Stokes exact solution u and p depending on the mesh size



(a) The error of the velocity u

(b) The error of the pressure p

Figure 3.4: Error analysis for Stokes exact solution

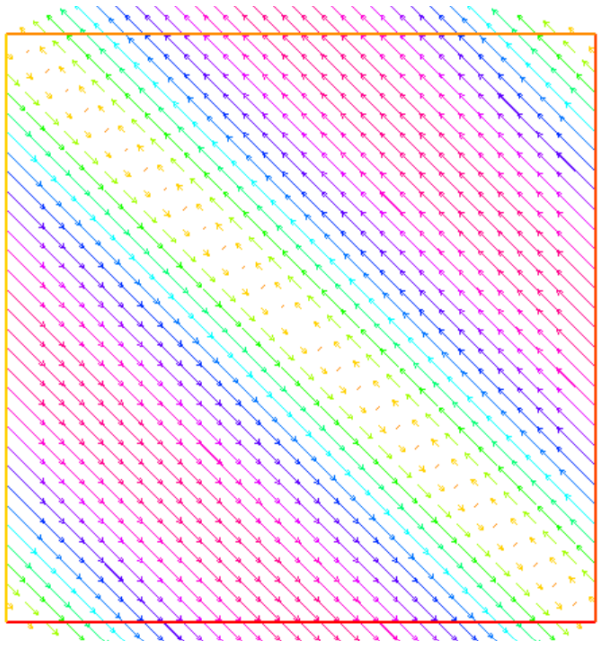
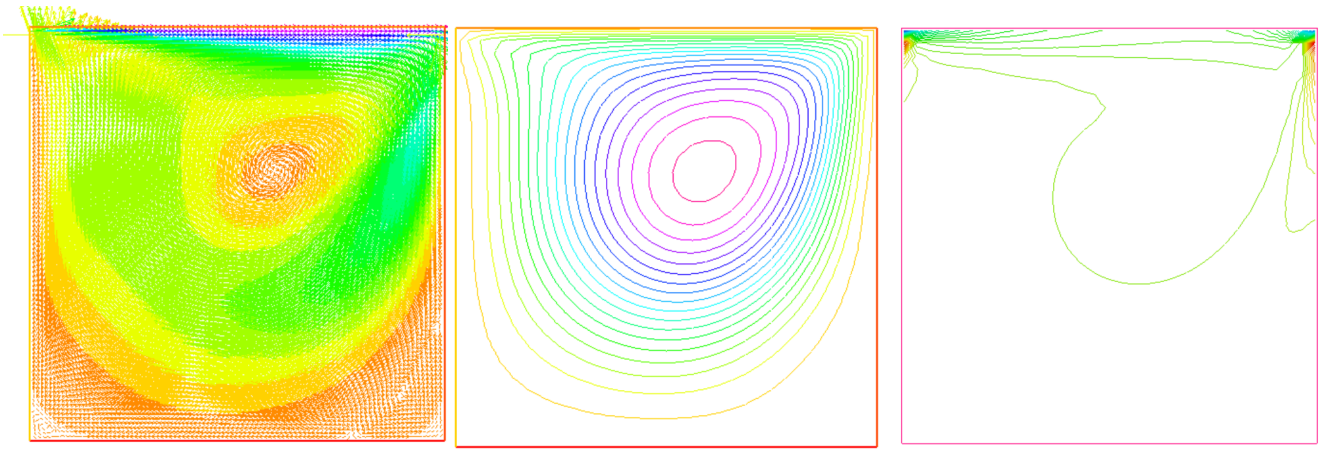


Figure 3.5: Fluid velocity under Stokes equations with known solution

3.2 Simulations of Navier-Stokes problem

3.2.1 The cavity test

The geometry of the cavity and the boundary conditions in this simulation are identical to those of the cavity test using the Stokes equations, as defined in section 3.1.1 and illustrated in Figure 3.1. The



(a) Velocity (b) Streamlines (c) Vorticity

Figure 3.6: Driven cavity flow at $Re = 400$

formulation of the Navier-Stokes problem used in this simulation is :

$$\begin{cases} \dfrac{Du}{Dt} - \dfrac{1}{Re}\Delta u + \nabla p = 0 \text{ in } \Omega, \\ \nabla . u = 0 \text{ in } \Omega. \end{cases} \tag{3.4}$$

The Reynolds number is defined by :

$$Re = \frac{u_0 D}{\nu}, \tag{3.5}$$

where u_0 is the first component of the velocity on the upper wall, and D is the size of the square cavity, which, in this case, is the unit square. Thus, in our case, $Re = \frac{1}{\nu}$.

The vorticity ω, which is the curl of the velocity u, is a valuable tool for visualising the vortices. The vorticity is defined by the following equation :

$$\omega = \frac{\partial u_2}{\partial x} - \frac{\partial u_1}{\partial y}. \tag{3.6}$$

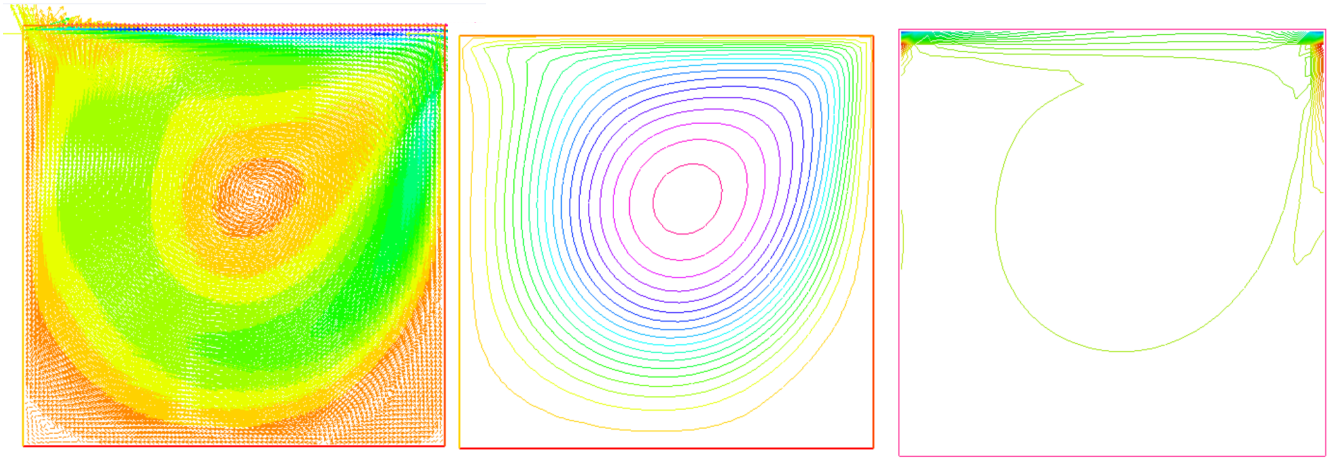
In the context of a two-dimensional problem, the Navier-Stokes equations can be formulated in an interesting manner by employing the stream function ψ. The stream function for the velocity field is defined by :

$$u_1 = \frac{\partial \psi}{\partial y}, \quad u_2 = -\frac{\partial \psi}{\partial x}. \tag{3.7}$$

For the stream formulation, the boundary conditions become : $\psi = 0$ on $\partial\Omega$.

The first simulation was executed using a mesh of 32×32 and a Reynolds number of $Re = 400$. Figure 3.6 illustrates the velocity, streamlines and vorticity of the flow within the driven cavity. Figure 3.7 corresponds to $Re = 1000$. The outcomes of these simulations are consistent with those obtained by [15], as well as those of [17].

In order to verify the program once more, a comparison of the position of the centre of the main vortex is necessary. For this purpose, a Reynolds number $Re = 1000$ is utilised, and a comparison of various meshes is conducted. Table 3.2 presents the position of the centre for different meshes. The table also shows the results of other studies for comparison, such as in [12]. It is apparent that our results are not as precise as the ones obtained by the referenced studies. This can be attributed to the utilisation of a more simplified methodology and the utilisation of the FreeFem++ software. However, as the mesh becomes finer, the results approach those obtained by the other studies.



(a) Velocity (b) Streamlines (c) Vorticity

Figure 3.7: Driven cavity flow at $Re = 1000$

Reference	Mesh	x	y
Present	50×50	0.54	0.60
Present	70×70	0.542857	0.6
Present	96×96	0.541667	0.583333
Present	141×141	0.539007	0.58156
[6]	96×96	0.4692	0.5652
[6]	128×128	0.4692	0.5652
[23]	141×141	0.52857	0.56429
[12]	401×401	0.53	0.565

Table 3.2: Positions of the main vortex

3.2.2 The Taylor vortex problem

In order to test and validate the program, a simulation of the Taylor vortex problem was conducted. This problem is an exact solution of the Navier-Stokes equations : find $u \in V$ and $p \in Q$ such that

$$\begin{cases} \dfrac{Du}{Dt} - \nu\Delta u + \nabla p = 0 \text{ in } \Omega, \\ \nabla . u = 0 \text{ in } \Omega, \\ u = g \text{ on } \partial\Omega, \end{cases} \tag{3.8}$$

where $g = (g_1, g_2)^T$ is defined by :

$$g_1(x,y) = -\pi \cos(\pi x)\sin(\pi y)\exp(\frac{-2\pi^2 t}{Re}),$$
$$g_2(x,y) = \pi \sin(\pi x)\cos(\pi y)\exp(\frac{-2\pi^2 t}{Re}).$$

The problem (3.8) admits the exact solution :

$$\begin{cases} u_1(x,y) = -\pi \cos(\pi x)\sin(\pi y)\exp(\dfrac{-2\pi^2 t}{Re}), \\ u_2(x,y) = \pi \sin(\pi x)\cos(\pi y)\exp(\dfrac{-2\pi^2 t}{Re}), \\ p(x,y) = -\dfrac{1}{4}\pi(\cos(2\pi x) + \cos(2\pi y))\exp(\dfrac{-4\pi^2 t}{Re}). \end{cases} \tag{3.9}$$

As the aim of this simulation is to validate the program, we have adopted the conditions set out in [27]. The domain is defined as $\Omega =]-\frac{1}{2}, \frac{3}{2}[\times]-\frac{1}{2}, \frac{3}{2}[$, the mesh size is 50×50 and the Reynolds number is $Re = 1$. The time interval is set at $\Delta t = 0.01$.

However, for this problem, the stream function ψ is mostly used to formulate the Navier-Stokes equations, as defined in (3.7). Consequently, to facilitate comparison with references [27] and [4], the streamlines of the solution will be calculated.

Figure 3.8 depicts the Taylor vortex at time $t = 0$, which corresponds to the initial condition of the simulation. The figure is consistent with the expected characteristics of a Taylor vortex and is similar to the one presented in [27].

The Taylor vortex is a decaying vortex, and thus it is possible to calculate its energy (3.10), for the purpose of verifying its consistency. It is expected that the energy will decrease with the progression of the time. Figure 3.9 clearly demonstrates the decay of the energy, which also validates the program.

$$\int_{-1/2}^{3/2}\int_{-1/2}^{3/2} (u_1^2 + u_2^2)\, dx\, dy \tag{3.10}$$

Finally, we plot the curves representing the error between the finite element solution and the theoretical solution (Figure 3.10). The error is calculated using the L^2 norm. Specifically, the following values are calculated : $\|u_1 - u_{1h}\|_{L^2(\Omega)}$, $\|u_2 - u_{2h}\|_{L^2(\Omega)}$ and $\|p - p_h\|_{L^2(\Omega)}$. As anticipated, the curves are observed to decrease.

3.3 Simulation of the advection equation

In order to achieve the objective of developing a simulation that incorporates both the Navier-Stokes and the advection equations, a preliminary simulation was conducted using only the advection equation. This was done in order to test the program and the discontinuous Galerkin formulation.

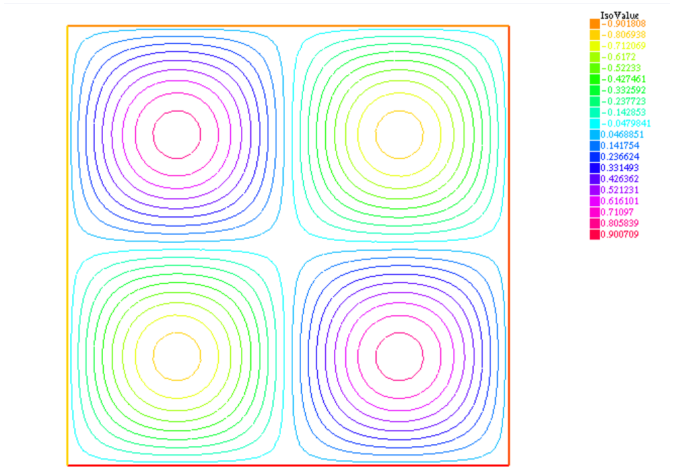


Figure 3.8: Taylor vortex streamlines at time $t = 0$

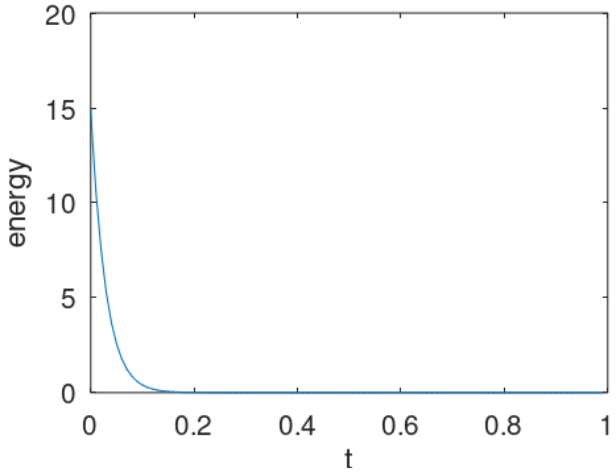
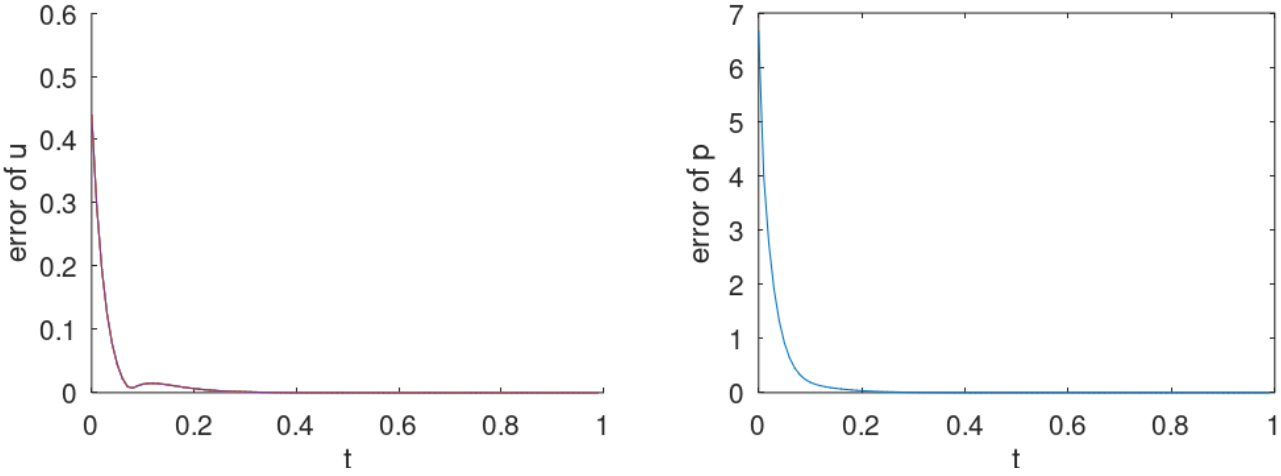


Figure 3.9: Energy of the Taylor vortex



(a) The error of the velocity u

(b) The error of the pressure p

Figure 3.10: Error analysis for Taylor vortex problem

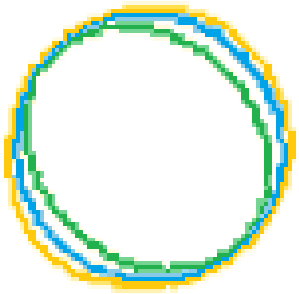


Figure 3.11: First position (yellow), final position for characteristics method (blue) and final position for discontinuous Galerkin method (green) of the circle transport by the advection equation

	method of characteristics	discontinuous Galerkin method
$\|u - u_h\|_{L^1(\Omega)}$	0.00650182	0.0205952

Table 3.3: Position error after a full rotation

The simulation model involves is a circle that is transported by the advection equation in $\Omega =]-1,1[\times]-1,1[$:

$$\begin{cases} \dfrac{\partial \phi}{\partial t} + u.\nabla\phi = 0, \\ \phi(t=0) = \phi_0, \end{cases} \tag{3.11}$$

where u is a rotating field such that $u = (y, -x)$ and $\phi_0 = \exp(-50((x-x_0)^2 + (y-y_0)^2))$.

We simulate a total rotation from $t = 0$ to $t = 2\pi$, with a time step of $\Delta t = \frac{2\pi}{1000}$ and a mesh size of 100×100. Two methods are utilised : the classic method of characteristics and the discontinuous Galerkin method. The final positions of the circle are compared to the initial position for both methods. As can be seen in Figure 3.11, the final circle produced by each method is almost identical to the initial circle in terms of both position and shape. Moreover, as shown in Table 3.3, the error produced by the discontinuous Galerkin method is sufficiently small for the program to be validated.

3.4 Simulations of a flow around an obstacle

We now aim to simulate a flow around an obstacle using the Navier-Stokes equations, and in a further time introduce pollution driven by the advection equation.

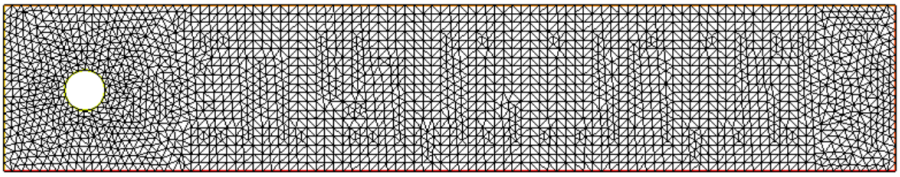
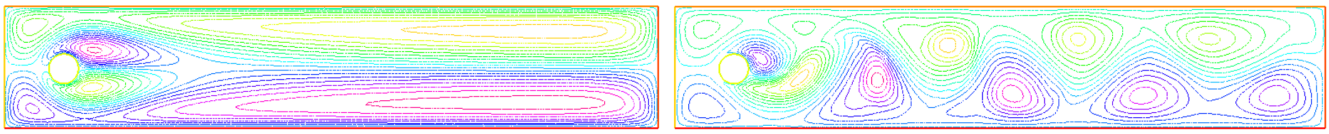


Figure 3.12: Mesh for the simulation of a flow around a circular cylinder



(a) Reynolds number = 100 (b) Reynolds number = 10000

Figure 3.13: Streamlines of the velocity around a circular cylinder

3.4.1 Simulations with the Navier-Stokes equations

First, we simulate a flow around a circular cylinder, with only the Navier-Stokes equations. The fluid enters through the left boundary (Γ_4) with a fixed velocity, and exits through the right boundary (Γ_2), which is a free-slip boundary. On the two other boundaries (Γ_1,Γ_3) the velocity is imposed equal to zero. Figure 3.12 illustrates the mesh for this simulation. We set the time step at $\Delta t = 0.01$. Readers may refer to [19] for a detailed discussion of the influence of mesh and time step on the solution.

Figure 3.13 depicts the streamlines of the velocity, for two distinct Reynolds numbers, $Re = 100$ and $Re = 10000$. It illustrates the impact of the Reynolds number on the shape of the vortices formed behind the obstacle, which is a significant point of consideration for the analysis of flow past an obstacle.

In a second time, the obstacle is change to a square cylinder and not a round cylinder. The geometry of the problem is shown in Figure 3.14, the dimension H and W are adjusted according to the desired simulations in order to compare the results with different studies. The time step is $\Delta t = 0.01$ and the mesh size is 80×20. Primary, the dimensions are set at $D = 1, H = 10D, W = 24D$ to compare our results to the one of [21]. However, alternative dimensions could have been selected, as evidenced by [25] and [24], who also obtained interesting results that readers may wish to explore.

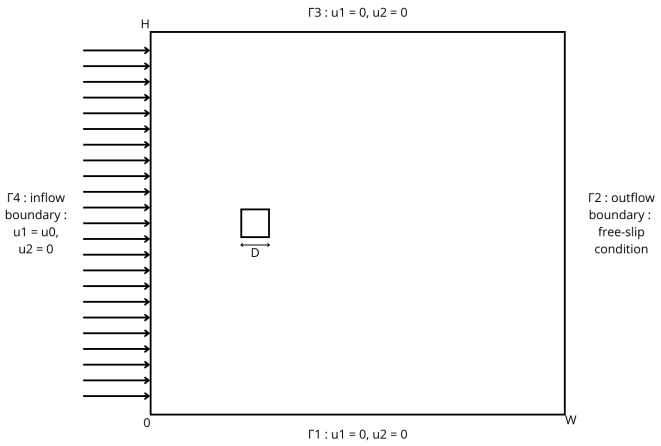
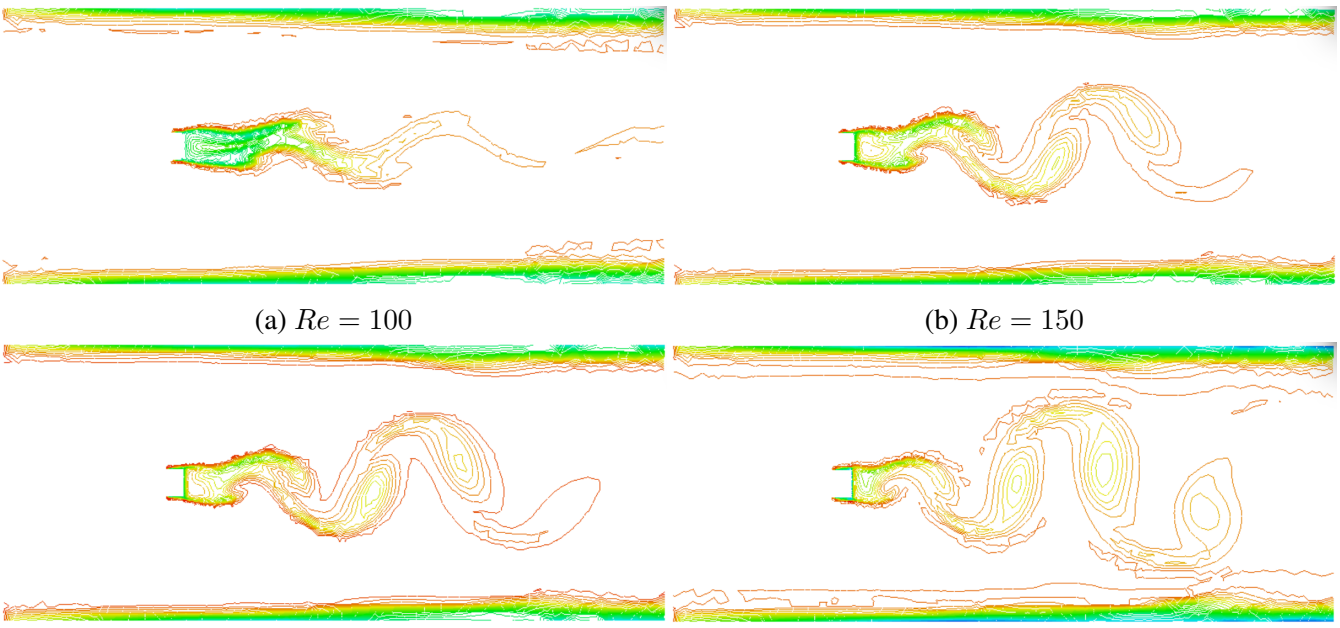


Figure 3.14: Geometry of the problem with a square cylinder



(a) $Re = 100$ (b) $Re = 150$

(c) $Re = 175$ (d) $Re = 250$

Figure 3.15: Velocity of the flow around a square cylinder

The Reynolds number is defined as :

$$Re = \frac{u_0 D}{\nu}, \tag{3.12}$$

where u_0 is the inflow velocity imposed on Γ_4, D is the width of the obstacle and ν is the kinematic viscosity.

Figure 3.15 illustrates the velocity of the fluid and the vortices that form as it passes the obstacle, at the same time, for Reynolds numbers between 100 and 250. As the Reynolds number increases, the vortices become more pronounced, and the wake behind the square cylinder becomes more complex. These results align with those obtained in [21].

3.4.2 Simulation with the Navier-Stokes equations and the advection equation

The objective is to integrate a pollution component into the simulation. The pollution will be driven by the advection equation mentioned in section 3.3. The domain is the same as the one mentioned above and defined in Figure 3.14, the dimensions are defined as follows : $D = 1, H = 13D, W = 14D$, in accordance with the specifications provided in [7]. The mesh size is 45×25, and the time step is set at $\Delta t = 0.02$. We will compare the vorticity between two Reynolds numbers, $Re = 1000$ and $Re = 1500$, see (3.6) for the vorticity equation.

Figure 3.16 illustrates the vorticity of the velocity field for the two distinct Reynolds numbers, $Re = 1000$ and $Re = 1500$, at the same iteration. As expected, the vorticity is more complex and considerably less uniform for the higher Reynolds number $Re = 1500$. In the case of $Re = 1000$, the flows emanating from both the top and bottom layers of the square cylinder are stable. Both flows roll up behind the obstacle and form the initial vortex. For $Re = 1500$, vortices appear in the flow coming by the top of the cylinder. Then they roll up downstream. These observations align with the findings presented in [7].

(a) $Re = 1000$ (b) $Re = 1500$

Figure 3.16: Vorticity of the flow around a square cylinder at two distinct Reynolds numbers

Conclusion

In conclusion, this work placement has been a great success. Indeed, regarding the subject, I have successfully simulated the Navier-Stokes equations using finite element methods, implemented with the FreeFem++ software. Validation of the simulations was conducted by implementing standard tests, such as the driven cavity test or the Taylor vortex problem. Additionally, error analyses were conducted to demonstrate the accuracy of the results. The results illustrated the sensitivity of the velocity flow to Reynolds number variations and demonstrated the potential benefits of combining the Navier-Stokes equations with other equations, such as the advection equation. The simulations also revealed the functioning of the finite element methods including the method of characteristics, the Galerkin method and the discontinuous Galerkin method, and their efficiency.

This internship enabled me to learn about finite element methods, including some advanced techniques such as the discontinuous Galerkin method. Moreover, this internship has contributed to a deeper comprehension of the Navier-Stokes equations and the significance of numerical simulations in fluid dynamics. This is in continuation with the course entitled Mathematical Modelling, taught by M. Le Roux, which I have taken in the third year of my major in Applied Mathematics and Modelling. In addition, I have gained competence in the use of the FreeFem++ software.

To further explore the subject of this internship, the implementation of three-dimensional simulations could provide more precise results and more accurately reflect the reality of engineering and scientific problems.

However, this internship has also been a rich experience on a personal level. Despite the significant cultural differences between Hanoi and the cities in France, I have found the ability to adjust and adapt to life in Hanoi. I really enjoyed discovering Hanoi and the Vietnamese culture. Even though I didn't know anyone, and I spend most of my time alone, I have managed to meet a few people. This internship has also facilitated significant progress in my English abilities, in particular in written English, which was the domain in which I was least comfortable previously. Finally, this internship has allowed me to reflect on my orientation towards research and has reinforced my choice.

Bibliography

[1] URL: https://parc-eolien-en-mer-de-saint-nazaire.fr/travaux-a-terre/.

[2] Alejandro Allievi and Rodolfo Bermejo. "Finite element modified method of characteristics for the Navier-Stokes equations". In: *International Journal for Numerical Methods in Fluids* 32.4 (Feb. 29, 2000), pp. 439–463. ISSN: 0271-2091, 1097-0363. DOI: 10.1002/(SICI)1097-0363(20000229)32:4<439::AID-FLD946>3.0.CO;2-Y.

[3] M. Amara, E. Chacón Vera, and D. Trujillo. "Vorticity-velocity-pressure formulation for Stokes problem". In: *Mathematics of Computation* 73.248 (Oct. 27, 2003), pp. 1673–1697. ISSN: 0025-5718, 1088-6842. DOI: 10.1090/S0025-5718-03-01615-6.

[4] Dr Blanca Bermudez and Dr René Posadas. "The Taylor Vortex and the Driven Cavity Problems by the Stream Function-Vorticity Formulation". In: *10th International Conference on Heat Transfer, Fluid Mechanics and Thermodynamics*. 2014.

[5] Daniele Boffi, Franco Brezzi, and Michel Fortin. "Finite Elements for the Stokes Problem". In: *Mixed Finite Elements, Compatibility Conditions, and Applications*. Vol. 1939. Series Title: Lecture Notes in Mathematics. Springer Berlin Heidelberg, 2008, pp. 45–100. ISBN: 978-3-540-78314-5 978-3-540-78319-0. DOI: 10.1007/978-3-540-78319-0_2.

[6] O. Botella and R. Peyret. "Benchmark spectral results on the lid-driven cavity flow". In: *Computers & Fluids* 27.4 (May 1998), pp. 421–433. ISSN: 00457930. DOI: 10.1016/S0045-7930(98)00002-4.

[7] Christophe Brun and Thomas Goossens. "3D coherent vortices in the turbulent near wake of a square cylinder". In: *Comptes Rendus. Mécanique* 336.4 (Feb. 20, 2008). Publisher: Cellule MathDoc/Centre Mersenne, pp. 363–369. ISSN: 1873-7234. DOI: 10.1016/j.crme.2008.01.002.

[8] Daniele Antonio Di Pietro and Alexandre Ern. *Mathematical Aspects of Discontinuous Galerkin Methods*. Vol. 69. Mathématiques et Applications. Springer Berlin Heidelberg, 2012. ISBN: 978-3-642-22979-4 978-3-642-22980-0. DOI: 10.1007/978-3-642-22980-0.

[9] Alexandre Ern and Jean-Luc Guermond. *Theory and Practice of Finite Elements*. Red. by S. S. Antman, J. E. Marsden, and L. Sirovich. Vol. 159. Applied Mathematical Sciences. Springer New York, 2004. ISBN: 978-1-4419-1918-2 978-1-4757-4355-5. DOI: 10.1007/978-1-4757-4355-5.

[10] Alexandre Ern and Jean-Luc Guermond. *Finite Elements I: Approximation and Interpolation*. Vol. 72. Texts in Applied Mathematics. Springer International Publishing, 2021. ISBN: 978-3-030-56340-0 978-3-030-56341-7. DOI: 10.1007/978-3-030-56341-7.

[11] Alexandre Ern and Jean-Luc Guermond. *Finite Elements II: Galerkin Approximation, Elliptic and Mixed PDEs*. Vol. 73. Texts in Applied Mathematics. Springer International Publishing, 2021. ISBN: 978-3-030-56922-8 978-3-030-56923-5. DOI: 10.1007/978-3-030-56923-5.

[12] E. Erturk, T. C. Corke, and C. Gökçöl. "Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers". In: *International Journal for Numerical Methods in Fluids* 48.7 (July 10, 2005), pp. 747–774. ISSN: 0271-2091, 1097-0363. DOI: 10.1002/fld.953.

[13] G.P. Galdi. *An Introduction to the Mathematical Theory of the Navier-Stokes Equations: Steady-State Problems*. Springer Monographs in Mathematics. Springer New York, 2011. ISBN: 978-0-387-09619-3 978-0-387-09620-9. DOI: 10.1007/978-0-387-09620-9.

[14] Vivette Girault and Pierre-Arnaud Raviart. *Finite Element Methods for Navier-Stokes Equations*. Vol. 5. Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 1986. ISBN: 978-3-642-64888-5 978-3-642-61623-5. DOI: 10.1007/978-3-642-61623-5.

[15] F. Glaisner and T.E. Tezduyar. *Finite element techniques for the Navier-Stokes equations in the primitive variable formulation and the vorticity stream-function formulation*. Tech. rep. NASA Contract NAS 9-17380, 1987.

[16] Frederic Hecht. *FreeFEM Documentation, Release 4.13*. June 13, 2024.

[17] Thomas J.R. Hughes, Wing Kam Liu, and Alec Brooks. "Finite element analysis of incompressible viscous flows by the penalty function formulation". In: *Journal of Computational Physics* 30.1 (Jan. 1979), pp. 1–60. ISSN: 00219991. DOI: 10.1016/0021-9991(79)90086-X.

[18] Jean-Baptiste APOUNG Kamga. *Stokes Equations. Implement in FreeFem++*. 2017.

[19] Hélène Persillon and Marianna Braza. "Physical analysis of the transition to turbulence in the wake of a circular cylinder by three-dimensional Navier–Stokes simulation". In: *Journal of Fluid Mechanics* 365 (June 25, 1998), pp. 23–88. ISSN: 00221120. DOI: 10.1017/S0022112098001116.

[20] O. Pironneau. "On the transport-diffusion algorithm and its applications to the Navier-Stokes equations". In: *Numerische Mathematik* 38.3 (Oct. 1982), pp. 309–332. ISSN: 0029-599X, 0945-3245. DOI: 10.1007/BF01396435.

[21] A.K Saha, G Biswas, and K Muralidhar. "Three-dimensional study of flow past a square cylinder at low Reynolds numbers". In: *International Journal of Heat and Fluid Flow* 24.1 (Feb. 2003), pp. 54–66. ISSN: 0142727X. DOI: 10.1016/S0142-727X(02)00208-4.

[22] Pierre Saramito. *Complex fluid modeling*. Dec. 8, 2014.

[23] R Schreiber and H.B Keller. "Driven cavity flows by efficient numerical techniques". In: *Journal of Computational Physics* 49.2 (Feb. 1983), pp. 310–333. ISSN: 00219991. DOI: 10.1016/0021-9991(83)90129-8.

[24] Ahmad Sohankar, L. Davidson, and C. Norberg. "Large Eddy Simulation of Flow Past a Square Cylinder: Comparison of Different Subgrid Scale Models". In: *Journal of Fluids Engineering* 122.1 (Mar. 1, 2000), pp. 39–47. ISSN: 0098-2202, 1528-901X. DOI: 10.1115/1.483224.

[25] Ahmad Sohankar, C. Norberg, and L. Davidson. "Simulation of three-dimensional flow around a square cylinder at moderate Reynolds numbers". In: *Physics of Fluids* 11.2 (Feb. 1, 1999), pp. 288–306. ISSN: 1070-6631, 1089-7666. DOI: 10.1063/1.869879.

[26] Atsushi Suzuki. *Introduction to finite element computation by FreeFem++ – towards numerical simulation of fluid flow problems*. Oct. 15, 2015.

[27] Daniel T. Valentine and A. Gaber Mohamed. "Taylor's Vortex Array: A New Test Problem for Navier-Stokes Solution Procedures". In: *Solution of Superlarge Problems in Computational Mechanics*. Ed. by James H. Kane, Arthur D. Carlson, and Donald L. Cox. Springer US, 1989, pp. 167–181. ISBN: 978-1-4612-7854-2 978-1-4613-0535-4. DOI: 10.1007/978-1-4613-0535-4_10.

Appendices

A Backgrounds

A.1 Functional spaces
A.2 Banach-Necas-Babuska problem

B FreeFem++ scripts

B.1 Program for the cavity test of Stokes
B.2 Program for the exact solution of Stokes
B.3 Program for the cavity test of Navier-Stokes
B.4 Program for the Taylor vortex problem
B.5 Program for the advection equation
B.6 Program for a flow around a circular cylinder
B.7 Program for a flow around a square cylinder

Appendix A

Backgrounds

In this appendix, a number of important backgrounds for finite element methods are provided. Initially, definitions of several significant function spaces are provided. Following this, a general definition of the Banach-Necas-Babuska (BNB) theorem is given, which is utilised to demonstrate the well-posedness of the problems under consideration. For a more detailed exposition, readers are referred to [9].

A.1 Functional spaces

Definition 1. (Banach space) *A Banach space is a complete normed vector space, which means that it is a vector space V equipped with a norm $\|.\|_V$ such that every Cauchy sequence in V has a limit in V.*

Definition 2. (Dual space) *V is a normed vector space, the dual space of V is $V' = \mathcal{L}(V;\mathbb{R})$.*

V is a Banach space, let's define the linear mapping $J_V : V \to V''$ by

$$\forall u \in V,\ \forall v' \in V',\ \langle J_V u, v' \rangle_{V'',V'} = \langle v', u \rangle_{V',V}.$$

Definition 3. (Reflexive Banach space) *V is a Banach space, then V is reflexive if J_V is an isomorphism.*

Definition 4. (Hilbert space) *A Hilbert space is an inner product space completed with respect to the norm defined by the inner product, and so is a Banach space.*

Definition 5. ([Inner product) *An inner product, also called a scalar product, on V is a bilinear mapping*

$$(.,.)_V : V \times V \ni (v,w) \mapsto (v,w)_V \in \mathbb{R},$$

satisfying three conditions :

1. *symmetry :* $\forall v, w \in V, (v,w)_V = (w,v)_V$,
2. *positivity :* $\forall v \in V, (v,v)_V \geq 0$,
3. $(v,v)_V = 0 \Leftrightarrow v = 0$.

Definition 6. *Let V and W be two normed vector spaces. $\mathcal{L}(V;W)$ is the vector space of continuous linear mappings from V to W.*

Definition 7. (Sobolev space) *Let s and p be two integers such that $s \geq 0$ and $1 \leq p \leq +\infty$. The Sobolev space $W^{s,p}(\Omega)$ is then defined as follows :*

$$W^{s,p}(\Omega) = \{u \in \mathcal{D}'(\Omega);\ \partial^\alpha u \in L^p(\Omega),\ |\alpha| \leq s\}.$$

The derivatives are understood in the distribution sense.

Proposition 8. *The space $W^{s,p}(\Omega)$ is a Banach space when it is equipped with the norm :*

$$\|u\|_{W^{s,p}(\Omega)} = \sum_{|\alpha| \leq s} \|\partial^\alpha u\|_{L^p(\Omega)}.$$

Theorem 9. *When $p = 2$ the space $W^{s,2}(\Omega)$ is a Hilbert space, if it is equipped with the scalar product :*

$$(u, v)_{s,\Omega} = \sum_{|\alpha| \leq s} \int_\Omega \partial^\alpha u\, \partial^\alpha v\, dx.$$

The space $W^{s,2}(\Omega)$ will then be denoted by $H^s(\Omega)$. The associated norm is denoted by $\|.\|_{s,\Omega}$.

A.2 Banach-Necas-Babuska problem

The following abstract problem is to be considered in its weak form :

$$\begin{cases} \text{find } u \in W \text{ such that } : \\ a(u, v) = (f, v) \quad \forall v \in V. \end{cases} \tag{A.1}$$

Where $a : W \times V \to \mathbb{R}$ is a bounded continuous bilinear form and $f : V \to \mathbb{R}$ is a continuous linear form.

Definition 10. *The problem (A.1) is well-posed if it admits a unique solution and :*

$$\exists c > 0, \quad \forall f \in V', \quad \|u\|_W \leq c\|f\|_{V'}. \tag{A.2}$$

The initial case to be considered is where W = V, in which case the problem can be stated as follows :

$$\begin{cases} \text{find } u \in V \text{ such that } : \\ a(u, v) = (f, v) \quad \forall v \in V. \end{cases} \tag{A.3}$$

In the context of V being a Hilbert space, the Lax-Milgram lemma can be employed to demonstrate the well-posedness of the problem, this is a necessary condition.

Lemma 11. (Lax-Milgram) *V is a Hilbert space, $a \in \mathcal{L}(V \times V; \mathbb{R})$ and $f \in V'$. Assume that the bilinear form a is coercive i.e. :*

$$\exists \alpha > 0, \quad \forall u \in V,\ a(u, u) \geq \alpha\|u\|_V^2. \tag{A.4}$$

Then the problem (A.3) is well-posed with :

$$\forall f \in V', \quad \|u\|_V \leq \frac{1}{\alpha}\|f\|_{V'}. \tag{A.5}$$

When considering a more general case, i.e. when W is a Banach space and V is a reflexive Banach space, the BNB theorem can be utilised to prove the well-posedness of the problem (A.1). It is important to note that the theorem is both a sufficient and a necessary condition.

Theorem 12. (Banach-Necas-Babuska) *W is a Banach space, V is a reflexive Banach space,* $a \in \mathcal{L}(W \times V; \mathbb{R})$ *and* $f \in V'$*. Then (A.1) is well-posed if and only if both following conditions are satisfied :*

$$\text{(BNB1)} \quad \exists \alpha > 0, \quad \inf_{w \in W} \sup_{v \in V} \frac{a(w,v)}{\|w\|_W \|v\|_V} \geq \alpha,$$
$$\text{(BNB2)} \quad \forall v \in V, \quad (\forall w \in W,\, a(w,v) = 0) \Rightarrow (v = 0).$$

Moreover :

$$\forall f \in V', \quad \|u\|_W \leq \frac{1}{\alpha} \|f\|_{V'}. \tag{A.6}$$

Appendix B

FreeFem++ scripts

B.1 Program for the cavity test of Stokes

Section 3.1.1

```
// Parameters
int nn = 30;
func f1 = 0;
func f2 = 0;

// Mesh
mesh Th = square(nn, nn);

// Fespace
fespace Uh(Th, P2);
Uh u1, u2;
Uh v1, v2;
fespace Ph(Th, P1);
Ph p, q;
Ph psi,phi;

macro Grad(u) [dx(u),dy(u),dz(u)] //
macro div(u1,u2) (dx(u1)+dy(u2)) //

// Problem
solve stokes ([u1, u2, p], [v1, v2, q])
  = int2d(Th)(
    dx(u1)*dx(v1)
    + dy(u1)*dy(v1)
    + dx(u2)*dx(v2)
    + dy(u2)*dy(v2)
    - p*div(v1,v2)
    - 1e-10*p*q
  )
  - int2d(Th)(
    f1*v1
    + f2*v2
  )
  - int2d(Th)(
    q*div(u1,u2)
  )
  + on(1, 2, 4, u1=0, u2=0)
```

```
  + on(3, u1=1, u2=0)
;

// Plot
plot(Th, wait=1);
plot([u1, u2],p, value=1, wait=1);
```

B.2 Program for the exact solution of Stokes

Section 3.1.2

```
func f1 = 2*sin(pi*(x + y)) + (1/pi)*cos(pi*(x + y));
func f2 = -2*sin(pi*(x + y)) + (1/pi)*cos(pi*(x + y));
func solu1 = (1/(pi*pi))*sin(pi*(x+y));
func solu2 = -(1/(pi*pi))*sin(pi*(x+y));
func solp = (1/(pi*pi))*sin(pi*(x+y));

macro div(u1,u2) (dx(u1)+dy(u2)) //

int n,i ;
int nbite = 10;
real[int] hvals(nbite), erru1vals(nbite), erru2vals(nbite), errpvals(nbite);

for (i=0; i<nbite; i++){
  n = (i+1)*10;

  // Mesh
  mesh Th = square(n, n);

  // Fespace
  fespace Uh(Th, P1b);
  Uh u1, u2;
  Uh v1, v2;
  fespace Ph(Th, P1);
  Ph p, q;

  // Problem
  solve stokes ([u1, u2, p], [v1, v2, q])
    = int2d(Th)(
      dx(u1)*dx(v1)
      + dy(u1)*dy(v1)
      + dx(u2)*dx(v2)
      + dy(u2)*dy(v2)
      - p*div(v1,v2)
      - 1e-10*p*q
    )
    - int2d(Th)(
      f1*v1
      + f2*v2
    )
    - int2d(Th)(
      q*div(u1,u2)
    )
    + on(1, u1 = (1/(pi*pi))*sin(pi*(x)), u2 = -(1/(pi*pi))*sin(pi*(x)))
    + on(2, u1 = (1/(pi*pi))*sin(pi*(1+y)), u2 = -(1/(pi*pi))*sin(pi*(1+y)))
```

```
    + on(3, u1 = (1/(pi*pi))*sin(pi*(x+1)), u2 = -(1/(pi*pi))*sin(pi*(x+1)))
    + on(4, u1 = (1/(pi*pi))*sin(pi*(y)), u2 = -(1/(pi*pi))*sin(pi*(y)))
  ;

  real erru1 = sqrt(int2d(Th)((solu1 - u1)*(solu1 - u1)));
  real erru2 = sqrt(int2d(Th)((solu2 - u2)*(solu2 - u2)));
  real errp = sqrt(int2d(Th)((solp - p)*(solp - p)));
  real h = 1.0/n;

  hvals[i] = h;
  erru1vals[i] = erru1;
  erru2vals[i] = erru2;
  errpvals[i] = errp;

  plot([u1, u2], wait = 1);
}

cout<<"values of h : "<<hvals<<endl;
cout<<"values of the error of u1 : "<<erru1vals<<endl;
cout<<"values of the error of u2 : "<<erru2vals<<endl;
cout<<"values of the error of p : "<<errpvals<<endl;
```

B.3 Program for the cavity test of Navier-Stokes

Section 3.2.1

```
load "medit"
// Parameters
int nn = 201;
real dt=0.2, nbite=300;
real re=1000., nu=1.0/re;

// Mesh
mesh Th = square(nn, nn);

// Fespace
fespace Uh(Th, P2);
Uh u1, u2;
Uh v1, v2;
Uh u1old = 0, u2old = 0;
fespace Ph(Th, P1);
Ph p, q;
Ph psi,phi;
Ph vorticity;

macro Grad(u) [dx(u),dy(u),dz(u)] //
macro div(u1,u2) (dx(u1)+dy(u2)) //

problem NavierStokes ([u1,u2,p], [v1,v2,q], solver = Crout) =
  int2d(Th)(
    1.0/dt*(u1*v1+u2*v2)
    + nu*(Grad(u1)'*Grad(v1)+Grad(u2)'*Grad(v2))
    - 1e-10*p*q
    - p*div(v1, v2)
    - q*div(u1, u2)
```

```
  )
  - int2d(Th)(1.0/dt*(convect([u1old,u2old],-dt,u1old)*v1
                      +convect([u1old,u2old],-dt,u2old)*v2))
  + on(1, 2, 4, u1=0, u2=0)
  + on(3, u1=1, u2=0)
;

problem streamlines (psi, phi) =
  int2d(Th)(
    dx(psi)*dx(phi)
    + dy(psi)*dy(phi)
  )
  + int2d(Th)(
    - phi*(dy(u1) - dx(u2))
  )
  + on(1, 2, 4, 3, psi=0)
;

for(int i=0; i<nbite; i++){
  cout << i << endl;

  NavierStokes;
  streamlines;
  vorticity = (dx(u2) - dy(u1));

  u1old = u1;
  u2old = u2;

  savesol("Th." + i + ".sol", Th, [u1,u2]);
  savemesh(Th, "Th." + i + ".mesh");
}

real xcenter=0, ycenter=0;
real xtest=0,ytest=0;
real psimax=0;
for (int i = 0; i < Th.nv; i++) {
  xtest = Th(i).x;
  ytest = Th(i).y;
  if (psi(xtest,ytest)>psimax){
    psimax = psi(xtest,ytest);
    xcenter = xtest;
    ycenter = ytest;
  }
}

cout << "centre position : " << xcenter << ", " << ycenter << endl;
cout << "psi : " << psimax << endl;
```

B.4 Program for the Taylor vortex problem

Section 3.2.2

```
// Parameters
int nn = 50;
real re=1, nu=1.0/re;
```

```
real t = 0; //initial time
real T = 1.; //final time
real nbite = 100.;
real dt = T/(nbite);
int i = 0; //iteration

// Mesh
real x0 = -1./2., x1 = 3./2., y0 = -1./2., y1 = 3./2.;
mesh Th = square(nn, nn, [x0+(x1-x0)*x, y0+(y1-y0)*y]);

// Fespace
fespace Uh(Th, P1b);
Uh u1, u2;
Uh v1, v2;
Uh u1old = -pi*cos(pi*x)*sin(pi*y), u2old = pi*sin(pi*x)*cos(pi*y);
Uh u1oldtemp = convect([u1old,u2old],-dt,u1old);
Uh u2oldtemp = convect([u1old,u2old],-dt,u2old);
fespace Ph(Th, P1);
Ph p, q;
Ph psi, phi;

func solu1 = -pi*cos(pi*x)*sin(pi*y)*exp((-2*pi*pi*t)/re);
func solu2 = pi*sin(pi*x)*cos(pi*y)*exp((-2*pi*pi*t)/re);
func solp = (-1/4)*pi*(cos(2*pi*x)+cos(2*pi*y))*exp((-4*pi*pi*t)/re);
func f1 = 0;
func f2 = 0;

real[int] erru1vals(nbite), erru2vals(nbite), errpvals(nbite), energyvals(nbite);

macro Grad(u) [dx(u),dy(u)] //
macro div(u1,u2) (dx(u1)+dy(u2)) //

problem NavierStokes ([u1,u2,p], [v1,v2,q], solver = Crout) =
    int2d(Th)
    (
      1.0/dt*(u1*v1+u2*v2)
      + nu*(Grad(u1)'*Grad(v1)+Grad(u2)'*Grad(v2))
      - 1e-10*p*q
      - p*div(v1, v2)
      - q*div(u1, u2)
    )
    - int2d(Th)(1.0/dt*(u1oldtemp*v1+u2oldtemp*v2))
    - int2d(Th)(f1*v1 + f2*v2)
    + on(1, 3, u1 = pi*cos(pi*x)*exp((-2*pi*pi*t)/re), u2 = 0)
    + on(2, 4, u1 = 0, u2 = -pi*cos(pi*y)*exp((-2*pi*pi*t)/re))
;

problem streamlines (psi, phi) =
    int2d(Th)(
      dx(psi)*dx(phi)
      + dy(psi)*dy(phi)
    )
    + int2d(Th)(
      - phi*(dy(u1) - dx(u2))
    )
    + on(1, 2, 3, 4, psi=0)
```

```
;

while(t<T){
  func solu1 = -pi*cos(pi*x)*sin(pi*y)*exp((-2*pi*pi*t)/re);
  func solu2 = pi*sin(pi*x)*cos(pi*y)*exp((-2*pi*pi*t)/re);
  func solp = (-1/4)*pi*(cos(2*pi*x)+cos(2*pi*y))*exp((-4*pi*pi*t)/re);

  u1oldtemp=convect([u1old,u2old],-dt,u1old);
  u2oldtemp=convect([u1old,u2old],-dt,u2old);

  NavierStokes;
  streamlines;

  real energy = int2d(Th)(u1*u1+u2*u2);
  energyvals[i] = energy;

  real erru1 = sqrt(int2d(Th)((solu1 - u1)*(solu1 - u1)));
  real erru2 = sqrt(int2d(Th)((solu2 - u2)*(solu2 - u2)));
  real errp = sqrt(int2d(Th)((solp - p)*(solp - p)));
  erru1vals[i] = erru1;
  erru2vals[i] = erru2;
  errpvals[i] = errp;

  plot(psi, value = 1, fill = 0, cmm="iteration "+i+", time "+t);

  u1old = u1;
  u2old = u2;
  t += dt;
  i += 1;
}

cout<<"values of the error of u1 : "<<erru1vals<<endl;
cout<<"values of the error of u2 : "<<erru2vals<<endl;
cout<<"values of the error of p : " <<errpvals<<endl;
cout<<"values of the energy : " <<energyvals<<endl;
```

B.5 Program for the advection equation

Section 3.3

```
// Parameters
int n = 100, M = 10.*n;
real dt = 2*pi/M;

// Mesh
real x0 = 0.0, y0 = 0.5;
mesh Th = square(n,n, [x*2.-1.,y*2.-1.]);

// Fespace
fespace Vh(Th, P1);
Vh uconv = exp(-50*((x-x0)^2+(y-y0)^2)), vconv, uconvold;
Vh ufix = uconv;
Vh u1 = y, u2 = -x;
fespace Vhdg(Th, P1dc);
Vhdg udg = exp(-50*((x-x0)^2+(y-y0)^2)), vdg, udgold;
```

```
Vhdg ufixdg = udg;
Vhdg u1dg = y, u2dg = -x;

macro n() (N.x*u1dg + N.y*u2dg) //

problem Aconv(uconv,vconv)=
  int2d(Th)(uconv*vconv)
  - int2d(Th)(convect([u1,u2],-dt,uconvold)*vconv)
  + on(1,2,3,4, uconv = 0)
;

problem Adg(udg,vdg)=
  int2d(Th)(
    (udg/dt+(u1dg*dx(udg)+u2dg*dy(udg)))*vdg
  )
  + intalledges(Th)(
    (1-nTonEdge)*(abs(n)/2.)*jump(udg)*vdg
    -(1-nTonEdge)*n/2.*jump(udg)*vdg
  )
  - int2d(Th)(
    udgold*vdg/dt
  )
;

for (int m = 0; m < M; m++){
  uconvold = uconv;
  udgold = udg;
  Aconv;
  Adg;
  plot(ufix,uconv,udg, nbiso=3);
}

real erruconv = int2d(Th)(abs(ufix - uconv));
cout << "error of u with characteristics method : " << erruconv << endl;
real errudg =int2d(Th)(abs(ufixdg - udg));
cout << "error of u with dg method : " << errudg << endl;
```

B.6 Program for a flow around a circular cylinder

Section 3.4.1

```
load "medit"
// Mesh
real H = 0.41 , W = 2.2;
border c1(t=0,W){x=t; y=0;}
border c2(t=0,H){x=W; y=t;}
border c3(t=W,0){y=H; x=t;}
border c4(t=H,0){x=0; y=t;}
border C(t=0,2*pi){x=0.2+0.05*cos(t); y=0.2+0.05*sin(t);};
int n = 20;
mesh Th = buildmesh(c1(5*n) + c2(n) + c3(5*n) + c4(n) + C(-1*n));

// Fespace
fespace Uh(Th, P2);
Uh u1, u2;
```

```
15 Uh v1, v2;
16 Uh u1old, u2old;
17 Uh psi,phi;
18 fespace Ph(Th, P1);
19 Ph p, q;
20
21 // Parameters
22 real re = 10000, nu=1.0/re;
23 real dt=0.01, T = 10;
24 real um = 3;
25
26 macro Grad(u) [dx(u),dy(u),dz(u)] //
27 macro div(u1,u2) (dx(u1)+dy(u2)) //
28
29 problem NavierStokes ([u1,u2,p], [v1,v2,q]) =
30     int2d(Th)
31     (
32       1.0/dt*(u1*v1+u2*v2)
33       + nu*(Grad(u1)'*Grad(v1)+Grad(u2)'*Grad(v2))
34       - p*div(v1, v2)
35     )
36     - int2d(Th)(1.0/dt*(convect([u1old,u2old],-dt,u1old)*v1+convect([u1old,u2old],-
       dt,u2old)*v2))
37     + int2d(Th)(q*div(u1, u2))
38     - int2d(Th)(p*q*1e-10)
39     + on(c4, u1 = 4.*um*y*(0.41-y)/(0.41*0.41), u2 = 0)
40     + on(c1, c3, C, u1 = 0, u2 = 0)
41 ;
42
43 problem streamlines (psi, phi) =
44     int2d(Th)(
45       dx(psi)*dx(phi)
46       + dy(psi)*dy(phi)
47     )
48     + int2d(Th)(
49       - phi*(dy(u1) - dx(u2))
50     )
51     + on(c1, c2, c3, c4, C, psi=0)
52 ;
53
54 for(int i=0; i*dt<=T; i++){
55   cout << i << endl;
56
57   u1old = u1;
58   u2old = u2;
59
60   NavierStokes;
61   streamlines;
62
63   savesol("Th." + i + ".sol", Th, [u1,u2]);
64   savemesh(Th, "Th." + i + ".mesh");
65 }
66 plot(psi, wait = 1);
```

B.7 Program for a flow around a square cylinder

Section 3.4.2

```
load "medit"
// Mesh
real D=1, H=13*D, W = 14*D;
border fr1(t=0,W){x=t; y=0; label=1;}
border fr2(t=0,H){x=W; y=t; label=2;}
border fr3(t=W,0){x=t; y=H; label=3;}
border fr4(t=H,0){x=0; y=t; label=4;}
border in1(t =6*D,7*D){x = 3*D; y = t   ; label = 5;}
border in2(t =3*D,4*D){x = t   ; y = 7*D; label = 5;}
border in3(t =7*D,6*D){x = 4*D; y = t   ; label = 5;}
border in4(t =4*D,3*D){x = t   ; y = 6*D; label = 5;}
int nn=15;
int nx = 45;
int ny = 25;
mesh Th = buildmesh(fr1(nx)+fr2(ny)+fr3(nx)+fr4(ny)
                    +in1(nn)+in2(nn)+in3(nn)+in4(nn));

// Fespace
fespace Uh(Th, P2);
Uh u1, u2;
Uh v1, v2;
Uh u1old, u2old;
fespace Ph(Th, P1);
Ph psi,phi;
Ph p, q;
fespace Vh(Th, P1dc);
Vh cc = 0, w, ccold;
Vh beta1 = y, beta2 = -x;

// Parameters
real re = 1000;
real u0 = 10;
real nu = D*u0/re;
real t = 0;
real dt = 0.02, T = 10.;

macro Grad(u) [dx(u),dy(u),dz(u)] //
macro div(u1,u2) (dx(u1)+dy(u2)) //
macro n() (N.x*beta1+N.y*beta2) //

func real f(real a)
{
  if(a<0.01) return 1.;
  else return 0;
}
func real f2(real a)
{
  if(a>0.01) return cc;
  else return 1.;
}

problem NavierStokes ([u1,u2,p], [v1,v2,q], solver = Crout) =
    int2d(Th)(
```

```
      1.0/dt*(u1*v1+u2*v2)
      + nu*(Grad(u1)'*Grad(v1)+Grad(u2)'*Grad(v2))
      - p*div(v1, v2)
    )
    - int2d(Th)(
      1.0/dt*(convect([u1old,u2old],-dt,u1old)*v1
            + convect([u1old,u2old],-dt,u2old)*v2)
    )
    - int2d(Th)(q*div(u1, u2))
    - int2d(Th)(p*q*1e-10)
    + on(4, u1 = u0, u2 = 0)
    + on(1,3,5,u1 = 0, u2 = 0)
;

problem streamlines (psi, phi) =
    int2d(Th)(
      dx(psi)*dx(phi)
      + dy(psi)*dy(phi)
    )
    + int2d(Th)(
      - phi*(dy(u1) - dx(u2))
    )
    + on(1, 2, 3, 4, 5, psi=0)
;

problem Adual(cc,w) =
    int2d(Th)(
      (cc/dt+(beta1*dx(cc)+beta2*dy(cc)))*w
    )
    + intalledges(Th)(
      (1-nTonEdge)*(abs(n)/2.)*jump(cc)*w
      -(1-nTonEdge)*n/2.*jump(cc)*w
    )
    - int2d(Th)(
      ccold*w/dt
    )
;

u1 = 0;
u2 = 0;
p = 0;
cc = f(x);

for(int i=1; i<=T/dt; i++){
  t+=dt;
  cout << i << endl;

  u1old = u1;
  u2old = u2;

  NavierStokes;
  streamlines;
  Ph vorticity = dx(u2) - dy(u1);

  beta1 = u1;
  beta2 = u2;
```

```
110     ccold = f2(x);
111     Adual;
112
113     savesol("Th." + i + ".sol",Th,vorticity);
114     savemesh(Th, "Th." + i + ".mesh");
115 }
```

Numerical simulation of the Navier-Stokes equations using finite element methods

Abstract : The Navier-Stokes problem constitutes a compelling set of equations with numerous applications in diverse fields. The objective of this internship was to simulate these equations using finite element methods and the FreeFem++ software. The approach adopted involved the utilisation of the method of characteristics and the Galerkin method in order to solve the Navier-Stokes equations. The discontinuous Galerkin method was also studied, in order to solve the advection equation, a further important component of the numerical simulations. A series of simulations were conducted, with the most elementary being the driven cavity test case with the Stokes equations. The Taylor vortex problem was simulated to validate the program with the Navier-Stokes equations, and a simulation involving only the advection equation was conducted to validate the program implementing the discontinuous Galerkin method. Subsequently, simulations of a flow around an obstacle were conducted, with only the Navier-Stokes equations and then combining the Navier-Stokes equations and the advection equation. The outcomes demonstrate the efficacy of the proposed approaches in accurately capturing fluid behaviours in various contexts.

Keywords : Navier-Stokes equations, Advection equation, Method of characteristics, Galerkin method, Discontinuous Galerkin method, Numerical simulations, FreeFem++

Résumé : Le problème de Navier-Stokes constitue un fascinant ensemble d'équations, avec de nombreuses applications dans divers domaines. L'objectif de ce stage était de simuler ces équations à l'aide de méthodes des éléments finis et du logiciel FreeFem++. L'approche adoptée impliquait d'utiliser la méthode des caractéristiques et la méthode de Galerkin afin de résoudre les équations de Navier-Stokes. La méthode de Galerkin discontinue a également été étudié pour résoudre l'équation d'advection, une autre composante importante des simulations numériques. Une série de simulations a été réalisée, la plus élémentaire étant le cas test de la cavité entrainée avec l'équation de Stokes. Le problème du tourbillon de Taylor a été simulé pour valider le programme avec les équations de Navier-Stokes, et une simulation impliquant uniquement l'équation d'advection a été réalisée pour valider le programme mettant en oeuvre la méthode de Galerkin discontinue. Ensuite, des simulations d'un écoulement autour d'un obstacle ont été effectuées, avec seulement les équations de Navier-Stokes, puis en combinant les équations de Navier-Stokes et l'équation d'advection. Les résultats démontrent l'efficacité des approches proposées pour capturer avec précision les comportements des fluides dans divers contextes.

Mots-clefs : Equations de Navier-Stokes, Equation d'advection, Méthode des charactéristiques, Méthode de Galerkin, Méthode de Galerkin discontinue, Simulations numériques, FreeFem++