UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

BACHELOR FINAL THESIS

Study for the computational resolution of conservation equations of mass, momentum and energy. Numerical analysis and turbulence models

Document:

Report

Author:

Gerard Toset Alonso

Director/Co-director:

Carlos-David Pérez Segarra / Francesc Xavier Trias Miquel

Degree:

Bachelor in Aerospace Vehicle Engineering

Examination session:

Spring 2023.

Acknowledgements

En primer lloc, m'agradaria expressar el meu agraïment al professor Carlos-David Pérez Segarra per la motivació que em va proporcionar a la seva assignatura de *Dinàmica de Gasos i Transferència de Calor i Massa*, que em va introduir al món de l'enginyeria computacional aplicada a la mecànica de fluids. També li agraeixo tot el suport i l'ajuda que m'ha brindat durant aquest treball. Vull agrair també al CTTC pels recursos proporcionats, especialment al professor Francesc Xavier Trias Miquel per donar-me l'oportunitat d'assistir a les seves classes d'Enginyeria Computacional.

En segon lloc, vull expressar el meu sincer agraïment a la meva família, amics i al meu company de classe Pau Tolsau per tot el suport i encoratjament que m'han brindat. Sense ells, aquest camí no hauria estat tan planer com ho ha estat. Vull destacar especialment el suport i la comprensió de la meva parella, Janina Bonastre, que ha estat una font constant de motivació i ànims en tot moment. Estic profundament agraït a tots ells per haver estat al meu costat durant aquesta etapa.

Finalment, al meu iaio, que m'ha motivat a estudiar des de ben menut i no m'ha pogut veure acabant aquesta gran etapa. Gràcies.

Abstract

This undergraduate thesis presents a study on the numerical solution of the Navier-Stokes equations under various assumptions. To achieve this, several codes have been developed in C++ language to evaluate and verify the cases proposed by the Heat Transfer Technological Center (CTTC).

The thesis is divided into different chapters. Firstly, there is an introduction to numerical methods, where spatial and temporal discretizations are explained, along with algorithms for solving systems of equations. Then, the main body of the work consists of four chapters. The first chapter addresses the heat conduction phenomenon and solves a transient two-dimensional conduction case. The second chapter deals with the general convection-diffusion equation and applies it to four problems to validate the code. The third chapter implements the Fractional Step Method (FSM) and solves the Lid-Driven Cavity, Differential Heated Cavity, and Square Cylinder cases. The fourth chapter solves the Burgers' equation in the Fourier space. All four chapters include theoretical development, the algorithm's structure for solving the case, and one or several verification cases, along with their respective reference results.

Finally, there is a last chapter about the Turbulent Flow problem, where the extension to three dimensions of the numerical algorithm is explained.

Resum

Aquest treball de final de grau presenta l'estudi sobre la resolució numèrica de les equacions de Navier-Stokes sota diverses hipòtesis. Per fer-ho, s'han desenvolupat diversos codis en llenguatge C++ per tal d'avaluar i verificar els casos proposats pel Centre Tecnològic de Transferència de Calor (CTTC).

El treball es divideix en diferents capítols. Primerament hi ha una introducció als mètodes numèrics, on s'explica la discretització espacial, temporal, i els algoritmes de resolució de sistemes d'equacions. Després, entren els quatre capítols que són el cos principal del treball. El primer capítol tracta el fenòmen de conducció de calor i resol un cas de conducció en dos dimensions i estat transitori. El segon, tracta l'equació general de convecció-difusió i l'aplica a quatre problemes per verificar el codi. El tercer capítol implenta el Fractional Step Method i resol el cas de la Lid-Driven Cavity, la Differential Heated cavity i el cas del Square Cylinder, i el quart capítol resol l'equació de Burguers a l'espai de Fourier. Tots quatre capítols inclouen un desenvolupament teòric, l'estuctura de l'algoritme a seguir per resoldre el cas, i un o diversos casos de verificació, amb els respectius resultats de referència.

Finalment, hi ha un últim capítol sobre el problema del Flux Turbulent, on s'explica l'extensió l'esquema numèric a tres dimensions.

Contents

1 Introduction **1**

1.1 Object . . . 1
1.2 Scope . . . 1
1.3 Requirements . . . 2
1.4 Rationale . . . 2

2 State of the art **3**

2.1 The FVM . . . 4
2.1.1 Spatial discretization . . . 4
2.1.2 Temporal discretization . . . 6
2.1.3 Solvers . . . 7

3 Conduction heat transfer **9**

3.1 Mathematical formulation . . . 9
3.2 Code structure . . . 11
3.3 Code verification . . . 12
3.3.1 Case 1 . . . 12
3.3.2 Case 2 . . . 13
3.3.3 Case 3 . . . 16
3.3.4 Case 4 . . . 17
3.3.5 4-materials 3D rod . . . 19
3.3.6 Mesh size influence . . . 21
3.4 Discussion . . . 22

4 Convection and Diffusion **23**

4.1 The Convection-Diffusion Equation . . . 23
4.1.1 Central-Difference scheme - CDS . . . 24
4.1.2 Upwind-Difference scheme - UDS . . . 25
4.1.3 Exponential-Difference scheme - EDS . . . 26
4.1.4 Second-order Upwind linear extrapolation - SUDS . . . 27

4.1.5 Quadratic Interpolation for Convective Kinematics - QUICK . . . 28
4.2 Code structure . . . 29
4.3 Code verification . . . 29
4.3.1 Parallel flow . . . 30
4.3.2 Vertical flow . . . 32
4.3.3 Diagonal flow . . . 33
4.3.4 Smith-Hutton problem . . . 36
4.4 Discussion . . . 38

5 The Fractional Step Method 39
5.1 Mathematical formulation . . . 39
5.2 Staggered meshes . . . 41
5.2.1 The Checkerboard problem . . . 41
5.2.2 Staggered meshes in the FSM . . . 42
5.2.3 Time step choice . . . 44
5.3 Code structure . . . 45
5.4 Code verification . . . 46
5.4.1 Lid-Driven Cavity . . . 46
5.4.2 Differential Heated Cavity . . . 55
5.4.3 Flow between Parallel Plates . . . 63
5.4.4 Square Cylinder . . . 66
5.5 Discussion . . . 73

6 Turbulence 74
6.1 Introduction to Turbulence . . . 74
6.2 Burgers' equation in Fourier space . . . 76
6.2.1 LES model . . . 78
6.2.2 Burgers' equation resolution . . . 78
6.3 Code structure . . . 79
6.4 Numerical results . . . 80
6.4.1 Code verification. Re=40 . . . 80
6.4.2 Extension to Re=200 . . . 81

7 Turbulent Flow between Parallel Plates 83
7.1 Problem definition . . . 83
7.2 3D extension . . . 85
7.3 Future work . . . 86

8 Budget summary 88

9 Analysis and assessment of environmental and social implications 89

10 Conclusions **90**

10.1 Conclusions 90

10.2 Future actions 91

Bibliography **92**

List of Figures

2.1 High-Performance Computing - Flow past a circular cylinder from critical to trans-critical Reynolds numbers. Extracted from [2] . . . 3
2.2 Different types of mesh . . . 5
2.3 *Block-off Method* representation . . . 6

3.1 2D spatial discretization . . . 10
3.2 Verification case 1. Geometry and properties . . . 13
3.3 Verification case 2. Geometry and properties . . . 14
3.4 2^{nd} verification case results. Comparison between analytical and numerical solutions . . . 15
3.5 Verification case 3. Geometry and properties . . . 16
3.6 3^{rd} verification case results. Comparison between analytical and numerical solutions . . . 17
3.7 Verification case 4. Geometry and properties . . . 17
3.8 4^{th} verification case results. Comparison between analytical and numerical solutions . . . 19
3.9 Materials distribution. Extracted from [7] . . . 19
3.10 Temperature map after 5000 seconds and 50×50 mesh . . . 21
3.11 Temperature map after 5000 seconds for different meshes . . . 22

4.1 CDS scheme . . . 24
4.2 UDS scheme . . . 25
4.3 SUDS scheme . . . 27
4.4 QUICK scheme . . . 28
4.5 Parallel flow scheme. Geometry and boundary conditions . . . 30
4.6 Analytic solution for different Péclet numbers and 100×100 mesh . . . 31
4.7 Comparison between analytical and numerical results for EDS . . . 32
4.8 Vertical flow scheme. Geometry and boundary conditions . . . 32
4.9 Analytic solution with 100×100 mesh . . . 33
4.10 Comparison between analytical and numerical results for EDS, UDS and CDS . . . 33
4.11 Diagonal flow scheme. Geometry and boundary conditions . . . 34
4.12 Numerical solution for different Péclet numbers and 100×100 mesh with UDS . . . 35
4.13 Numerical solution for $Pe = \infty$ ($Pe = 10^7$) with finer meshes and UDS . . . 35

4.14 Smith-Hutton flow scheme. Geometry. . . . 36
4.15 Numerical solution for different ρ/Γ values and 100×100 mesh with UDS 37
4.16 Comparison between provided and obtained solutions for different ρ/Γ values and 100×100 mesh 38

5.1 Staggered mesh 41
5.2 Distances and values evaluation positions 43
5.3 Pressure evaluation positions 44
5.4 Lid-Driven Cavity scheme. Geometry and boundary conditions 46
5.5 Numerical solutions for Re=100 48
5.6 Numerical solutions for Re=400 49
5.7 Numerical solutions for Re=1000 50
5.8 Numerical solutions for Re=3200 51
5.9 Numerical solutions for Re=5000 52
5.10 Numerical solutions for Re=7500 53
5.11 Numerical solutions for Re=10000 54
5.12 Differential Heated Cavity scheme. Geometry and boundary conditions 55
5.13 Numerical solutions for Ra=10^3 58
5.14 Numerical solutions for Ra=10^4 59
5.15 Numerical solutions for Ra=10^5 60
5.16 Numerical solutions for Ra=10^6 61
5.17 Flow between Parallel Plates scheme. Geometry and boundary conditions. Case 1 63
5.18 Numerical solutions for Case 1 64
5.19 Flow between parallel plates scheme. Geometry and boundary conditions. Cases 2 and 3 . . . 65
5.20 Numerical solutions for Case 2 65
5.21 Numerical solutions for Case 3 66
5.22 Numerical solutions for Case 3 - Vertical 66
5.23 Square Cylinder scheme. Geometry and boundary conditions. 67
5.24 Velocity map for different Reynolds. Steady solution 68
5.25 Pressure map for different Reynolds. Steady solution 69
5.26 Velocity map for different Reynolds. Unsteady solution 70
5.27 Pressure map for different Reynolds. Unsteady solution 71
5.28 Streamwise and cross-stream velocities for Re=100. Reference and numerical solution comparison. 72
5.29 Streamwise and cross-stream velocities for Re=100 and different x positions. Reference and numerical solution comparison. 73

6.1 Energy cascade 75
6.2 Representation of all possible triadic interactions between modes. Extracted from [17] 79
6.3 Energy spectrum of the steady-state solution of the Burgers equation for Re=40 80
6.4 Energy spectrum of the steady-state solution of the Burgers equation. Extracted from [17] . . 80
6.5 Energy spectrum of the steady-state solution of the Burgers equation for Re=200 81

7.1 Turbulent Flow between Parallel Plates scheme. Geometry and boundary conditions. 85

List of Tables

3.1 1^{st} verification case results. Comparison between analytical and numerical solutions 13
3.2 Thermophysical properties of the materials 20
3.3 Specific boundary conditions 20

4.1 Specific values of ϕ, Γ, and S_ϕ 24
4.2 Expected results for the Smith-Hutton problem. Extracted from [8] 37

5.1 Computational and time convergence for the different simulations 54
5.2 Results comparison for Ra=10^3 and Ra=10^4. Extracted from [13] 62
5.3 Results comparison for Ra=10^5 and Ra=10^6. Extracted from [13] 62

7.1 Geometry of the problem. Extracted from [19] 84

8.1 Total budget of the thesis 88

Glossary

The next list describes several symbols and abbreviations that will be later used within the body of the document.

Abbreviations

2D, 3D: Two and three dimensions

CDS: Central-Difference Scheme

CFD: Computational Fluid Dynamics

CFL: Courant-Friedrich-Levy

CTTC: Centre Tecnològic de Transferència de Calor

CV: Control Volume

DNS: Direct Numerical Simulations

EDS: Exponential-Difference Scheme

FDM: Finite Difference Method

FEM: Finite Element Method

FFT: Fast Fourier Transform

FSM: Fractional Step Method

FVM: Finite Volume Method

HRS: High Resolution Scheme

LES: Large Eddy Simulations

NS: Navier-Stokes

QUICK: Quadratic Interpolation for Convective Kinematics

RANS: Reynolds Averaged Navier-Stokes Simulations

SUDS: Second-order Upwind Difference Scheme

TDMA: Tri-Diagonal Matrix Algorithm

UDS: Upwind-Difference Scheme

Symbols

α**:** heat transfer coefficient $[W/m^2 \cdot K]$

β**:** time integration coefficient

Γ**:** diffusion coefficient

δ**:** convergence parameter

δ_w**:** half-width for turbulent flow

ϵ**:** energy dissipation ratio $[J/s]$

η**:** Kolmogorov microscale

κ**:** volumetric expansion coefficient $[1/K]$

λ**:** thermal conductivity $[W/m \cdot K]$

μ**:** viscosity $[Pa \cdot s]$

ν**:** kinematic viscosity $[m^2/s]$

ρ**:** density $[kg/m^3]$

$\vec{\tau}$**:** shear stress tensor $[N/m^2]$

ϕ**:** general convection-diffusion variable

$\Omega_{xP}, \Omega_{yP}, \Omega_{zP}$**:** staggered CV volumes $[m^3]$

A_f**:** CV surface $(f = e, w, n, s, f, b)$ $[m^2]$

a_F, b_P**:** coefficients $(F = E, W, N, S, P, F, B)$

C_k**:** Kolmogorov constant $[m^{3/4}/kg^{1/4} \cdot s^{1/2}]$

C_n**:** integration coefficient $(n = 1, 2...)$

c_P**:** specific heat at constant pressure $[J/kg \cdot K]$

c_v**:** specific heat at constant volume $[J/kg \cdot K]$

D_f**:** diffusive term $(f = e, w, n, s)$

d**:** distance $[m]$

E**:** energy $[J]$

e**:** wall width $[m]$

F**:** force term

f_f**:** convective term $(f = e, w, n, s)$

$\vec{g}$**:** gravity $[m/s^2]$

H**:** height $[m]$

k**:** wave number

L**:** total length $[m]$

L_r**:** dimensionless recirculation length

l**:** partial length $[m]$

M**:** vertical number of nodes

m**:** slope of the energy spectrum

$\dot{m}$**:** mass flux $[kg/s]$

N**:** horizontal number of nodes

Nu**:** Nusselt number

n**:** time coefficient

$\vec{n}$**:** surface vector

Pe**:** Péclet number

Pr**:** Prandtl number

p**:** pressure $[Pa]$

$\vec{q}$**:** heat flux $[W/m^2]$

R**:** FSM coefficient

Ra**:** Rayleigh number

Re**:** Reynolds number

Re_τ**:** Friction Reynolds number

S**:** generic surface $[m^2]$

S_ϕ**:** diffusion source term

T**:** temperature $[K]$

t**:** time $[s]$

u**:** x-velocity $[m/s]$

u_i**:** internal energy $[J]$

u_τ**:** shear velocity $[m/s]$

V**:** volume $[m^3]$

v**:** y-velocity $[m/s]$

$\vec{v}$**:** velocity vector $[m/s]$

$\vec{v}^P$**:** predictor velocity $[m/s]$

W**:** total width $[m]$

w**:** z-velocity $[m/s]$

x**:** horizontal coordinate $[m]$

y**:** vertical coordinate $[m]$

∞**:** infinity number

Chapter 1

Introduction

1.1 Object

This project aims to develop a set of codes capable of solving heat and mass transfer problems involving fluid dynamics. For this purpose, a study of different numerical methods and an in-depth study of the physics behind these problems and the mathematical formulation are performed. These problems are offered by the CTTC department and once its solution is obtained, a real particular case is proposed to be solved following the same methodology.

1.2 Scope

The study includes the following tasks:

1. **Heat conduction:** study of the transient conduction and development of a code to solve the case applied to a two-dimensional domain with different materials.
2. **The Convection-Diffusion equation:** study of the general convection-diffusion equation and development of a code to solve different benchmark problems with different schemes.
3. **Fractional Step Method:** study of the method and development of a code for solving different benchmark problems, such as the Lid-Driven Cavity, the Differential Heated Cavity, or the Square Cylinder.
4. **Turbulence:** study of the phenomena and development of a code for solving the Burgers' equation.
5. **Turbulence study:** following the same methodology as for the previous cases, solve a turbulent channel flow problem.

The language chosen to write the different codes is $C++$.
In any case, the study will include:

1. **Experimental studies:** every single study listed above will be solved by a code.

As high-level deliverables, the following documents will be drafted:

1. **Report:** will include the introduction, theoretical studies, case studies, and results, together with particular and global conclusions.
2. **Budget:** economic study of the project implementation.
3. **Annexes:** the codes and other relevant data will be delivered in the annexes.

1.3 Requirements

1. The codes to solve all the problems must be written in $C++$ language.
2. The codes must be developed with comments to ease their understanding.
3. The codes must be optimized to reduce the computational time.

1.4 Rationale

Since the advent of computers, engineering has evolved so that the first option to consider when designing anything is its simulation. Simulations are the closest approximation to the reality that allows for saving the costs of designing and manufacturing experiments. For this reason, it is absolutely necessary that nowadays, an engineer must know the basics of Computational Fluid Dynamics (CFD) and how to transform the theory of the phenomena to be studied in codes and simulations as close to reality as possible.

Undoubtedly, the study of heat transfer is of great importance in many engineering applications, since processes involving heat transfer are widely used in a great variety of devices and systems. For example, heat transfer is crucial in the study of heat dissipation in electrical components, in the design of internal combustion engines, in the design of chemical reactors, in the control of temperature in manufacturing processes, in the heating and cooling of buildings, and many other applications.

Knowledge of the main physics of heat transfer and its application in practice is essential for these processes. Numerical methods are an important tool to study and understand heat transfer processes. Through numerical simulation, it is possible to investigate heat transfer processes in different scenarios and conditions and optimize the design of devices and systems for higher efficiency.

In summary, this work on numerical methods applied to the study of heat transfer is important because it helps to understand and optimize heat transfer processes in a variety of benchmark problems. The study of conduction, convection-diffusion, and turbulence in particular is fundamental to this task, and numerical methods are a key tool for the study of these processes.

Chapter 2

State of the art

CFD and numerical methods have revolutionized the field of fluid dynamics by providing powerful tools to simulate and analyze complex fluid flow phenomena. With the advancement of computer technology and the development of sophisticated algorithms, CFD has become an indispensable tool in various industries, ranging from the energy and environmental sciences to aerospace and automotive engineering. Since 1960, CFD techniques have been integrated into the design and manufacture of airplane components and aircraft [1].

CFD involves the numerical solution of governing equations that describe fluid flow behavior. These equations, typically derived from the fundamental laws of physics, such as the conservation of mass, momentum, and energy, are solved using numerical methods to obtain approximate solutions. Numerical methods play a crucial role in CFD by discretizing the governing equations and transforming them into algebraic equations that can be solved using computational algorithms.

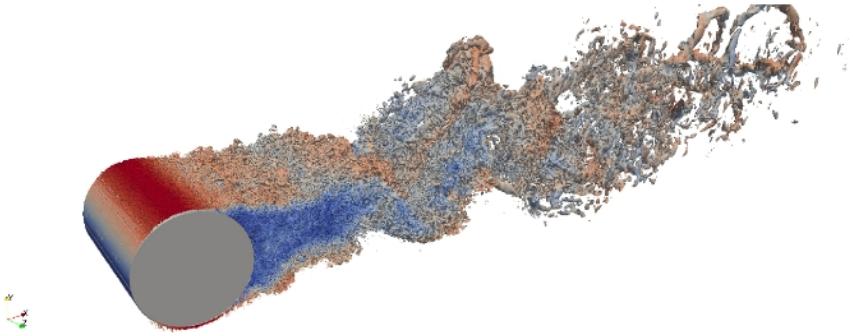


Figure 2.1: High-Performance Computing - Flow past a circular cylinder from critical to trans-critical Reynolds numbers. Extracted from [2]

The main scheme used throughout the work is based on the spatial and temporal discretization of the equations using the Finite Volume Method (FVM), transforming them into an algebraic system that is solved by means of an iterative process.

2.1 The FVM

Methods such as Finite Difference Method (FDM) or Finite Element Method (FEM) are also discretization methods that take the governing equations and discretize them into numerical systems that can be solved by computers. To give an example, FEM divides the domain into a finite number of subdomains called elements, and functions are applied to these elements. The whole domain function is defined by the summary of all the elements. This report is based on the FVM.

The FVM breaks the domain into control volumes (CV) or grid cells, defined by a grid. Then, nodes are placed at the cell faces or cell centers, and the flow properties are applied to these nodes [3]:

1. **Face-centered nodes:** nodes are placed at the center of the faces, which are determined by the grid. The centroid values are obtained by the mean values of the contiguous nodes.
2. **Cell-centered nodes:** nodes are placed at the CV centroid. The face values are interpolated with different schemes and the contiguous node values. It is considered that the whole CV has a constant value of a certain property, which is also known as *cell average* property value [4].

It is mandatory to create a mesh generation function in order to apply the FVM. The following subsection present the most commonly used grids nowadays.

2.1.1 Spatial discretization

As explained before, the FVM relies on a spatial discretization of the fluid domain to apply the numerical approximation of the solution. Down below are presented the main meshes that are used in the present. It must be kept in mind that every problem has its particular geometry, thus there is no correct or incorrect mesh.

The main division of meshes is the structured or unstructured meshes, and the hybrid ones [3]:

1. **Structured mesh:** the grid characteristics, such as cell faces or node positions can easily be identified by an index in every direction, and they are logically placed. To give an example, the third node of the first row is at the left of the fourth node of the same row, and it is above the third node of the second row. The structured meshes can also be sub-classified depending on the geometry of the CV (a and b), and their uniformity (c and d):
   (a) **Orthogonal mesh:** all the CVs have the same geometry and structure. It can also be defined as all the grid lines crossing each other by right angles (*see Figure (2.2a)*).
   (b) **Non-orthogonal mesh:** the geometry of the CVs is not constant along the mesh, meaning that the grid lines are crossing in angles different than 90°. This kind of mesh is widely used for circular geometries (*see Figure (2.2b)*).
   (c) **Uniform mesh:** the separation between grid lines is constant along the mesh.
   (d) **Non-uniform mesh:** the separation between grid lines is not constant. It is commonly increased in the zones where is expected to happen the major perturbations of the flow and a detailed view

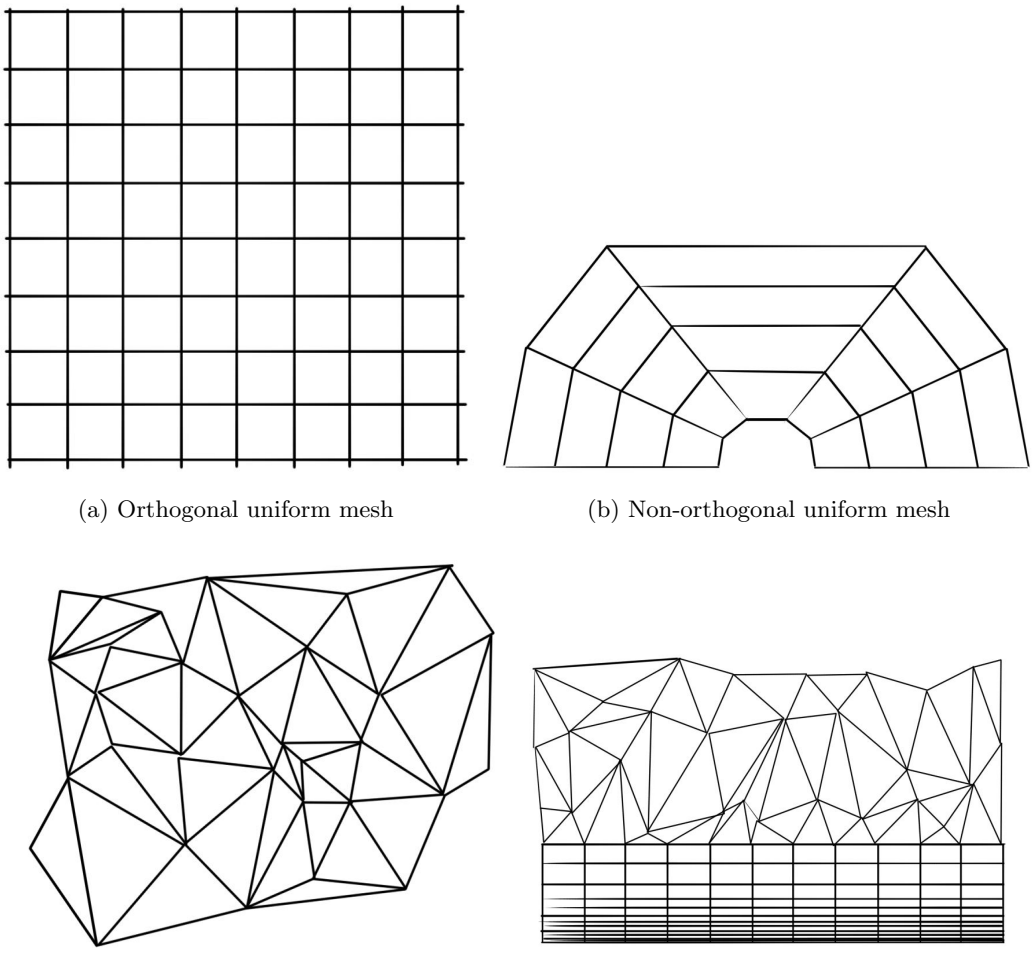
of the phenomenon wants to be captured.

Its efficiency and ease of definition make this kind of mesh a very suitable option for simulations. Notwithstanding, complex geometries can not be enough precisely defined with this type of grid. In other cases, the precision is certainly sacrificed and the *Block-off Method* is applied, which is explained below.

2. **Unstructured mesh:** the placement is no longer a logical function. For example, the first node can be at the right of the tenth node, and below the fifth one. Despite its computational cost increase, this mesh offers the capability to adapt to the most complex geometries (*see Figure (2.2c)*).

3. **Hybrid mesh:** this last case blends structured orthogonal non-uniform meshes near the walls of the geometry, with unstructured meshes where the flow is not expected to suffer huge variations and, thus, a detailed study of the zone is not needed (*see Figure (2.2d)*).

The following image sums up different mesh types.



(a) Orthogonal uniform mesh

(b) Non-orthogonal uniform mesh

(c) Unstructured mesh

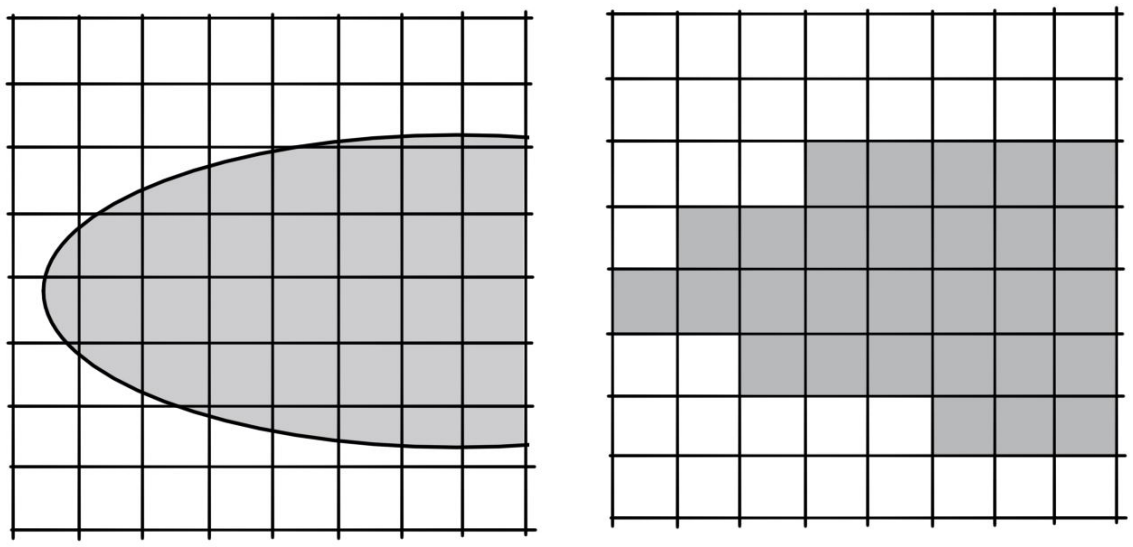
(d) Hybrid non-uniform mesh

Figure 2.2: Different types of mesh

Another important feature of spatial discretization and the meshes is where the values are stored. It has been

said that the values are stored at the nodes, but this is only true for collocated meshes. There is a special type of meshes in which scalar variables are stored at the nodes, but momentum variables are stored in the cell faces. These are known as staggered meshes.

The *Block-off Method* consists of a geometrical approximation of the fluid domain. It is common to study the flow around a curved surface. In this case, it is wanted to avoid the densification of the mesh and an orthogonal uniform mesh must be used. For this, the surface is no longer a perfect curve and it is approximated to a phased curve, as the picture shows:



(a) Real solid (b) Approximation of the solid

Figure 2.3: *Block-off Method* representation

2.1.2 Temporal discretization

For unsteady solutions, it is mandatory to consider the temporal variation of the flow. As well as geometry, time is also discretized into a finite number of time instants, and denoted by n for the current time instant, $n-1$ for the previous, and $n+1$ for the next one [3]. Considering a velocity vector $\vec{v}$, its time dependency can be expressed as follows:

$$\frac{\partial \vec{v}}{\partial t} = f(\vec{v}) \tag{2.1}$$

The time integration can be first-order discretized as the next expression shows:

$$\frac{\vec{v}^{n+1} - \vec{v}^{n}}{\Delta t} = f(\vec{v}^{n+1}, \vec{v}^{n}) \tag{2.2}$$

It is clearly visible that there is a dependency on the next and current velocity. The previous equation can be expressed in terms of a $\beta \in [0,1]$ parameter. The value of this parameter will determine the type of scheme chosen:

$$\frac{\vec{v}^{n+1} - \vec{v}^{n}}{\Delta t} = \beta f(\vec{v}^{n+1}) + (1-\beta) f(\vec{v}^{n}) \tag{2.3}$$

1. **Explicit scheme:** the solution of the next velocity only depends on the previous time instant value.

In other words, $\beta = 0$.

$$\vec{v}^{n+1} = \vec{v}^{n} + \Delta t f(\vec{v}^{n}) \tag{2.4}$$

This scheme is easy to program since it does not need a solver. The solution of the next time step only depends on the previous solution, which is already known. On the other hand, it presents stability problems, depending on the Δt.

2. **Implicit scheme:** the solution of the next velocity depends on the solution itself. The parameter is $\beta = 1$.

$$\vec{v}^{n+1} = \vec{v}^{n} + \Delta t f(\vec{v}^{n+1}) \tag{2.5}$$

This scheme is unconditionally stable, no matter the value of the time step. The main disadvantage is that a solver is mandatory. Some solvers are explained in the following section.

3. **Crank-Nicolson scheme:** this is also known as semi-implicit scheme since $\beta = 0.5$. The solution of the next time instant depends on both the current and next time instant. It is also unconditionally stable and it also needs a solver, but it is a second-order accurate scheme.

As well as spatial discretization, temporal discretization scheme selection relies on the problem itself. Depending on the situation and conditions, explicit or implicit schemes could present both advantages and disadvantages over the other scheme.

2.1.3 Solvers

As said before, for certain time integration schemes, solvers are needed. The following two solvers are presented down below [5]:

1. **Direct solver:** the most commonly used direct solver is the Tri-Diagonal Matrix Algorithm (TDMA). The following equation wants to be solved:

$$a_P(i)T(i) = a_E(i)T(i+1) + a_W(i)T(i-1) + b_P(i) \tag{2.6}$$

In matrices, it can be expressed as follows:

$$\begin{pmatrix} a_P(1) & a_E(1) & 0 & 0 & 0 & ... \\ -a_W(2) & a_P(2) & -a_E(2) & 0 & 0 & ... \\ 0 & -a_W(3) & a_P(3) & -a_E(3) & 0 & ... \\ ... & ... & ... & ... & ... & ... \end{pmatrix} \cdot \begin{pmatrix} T(1) \\ T(2) \\ T(3) \\ ... \end{pmatrix} = \begin{pmatrix} b_P(1) \\ b_P(2) \\ b_P(3) \\ ... \end{pmatrix} \tag{2.7}$$

The Tri-Diagonal matrix can be easily seen above. For 2D cases, it becomes Penta-Diagonal, and for

3D, Hepta-Diagonal. The equation is rewritten as follows, and substituted inside equation (2.6):

$$T(i-1) = P(i-1)T(i) + R(i-1) \tag{2.8}$$

$$a_P T(i) = a_E T(i+1) + a_W(P(i-1)T(i) + R(i-1)) + b_P(i) \tag{2.9}$$

This previous equation can be rewritten in terms of the next velocity. It must be seen that the TDMA solves from the last temperature instant to the previous one:

$$T(i) = \frac{a_E(i)}{a_P(i) - a_W(i)P(i-1)}T(i+1) + \frac{b_P(i) + a_W(i)R(i-1)}{a_P(i) - a_W(i)P(i-1)} \tag{2.10}$$

2. **Undirect solvers:** also known as iterative solvers, they take an initial estimated value T^{est} and a convergence parameter δ, and the iterative process continues until the convergence criterion is reached. The most commonly used direct solver is the Gauss-Seidel:

$$T(i) = \frac{a_E(i)T^{est}(i+1) + a_W(i)T^{est}(i-1) + b_P(i)}{a_P(i)} \tag{2.11}$$

If $|T^{est} - T(i)| > \delta$, there is a data update step, such as $T^{est} = T(i)$. If not, the convergence criterion has been reached.

Chapter 3

Conduction heat transfer

This third chapter is focused on the study and the appliance of the first thermal conduction phenomenon, heat conduction. For this purpose, a previous mathematical study is done to better understand the formulation and physics behind heat conduction, and after that, it is applied to different theoretical cases.

3.1 Mathematical formulation

The first approach to heat conduction can start with the energy conservation equation:

$$\frac{\partial}{\partial t}\int \rho u_i dV = -\int \vec{q}\cdot\vec{n}dS \tag{3.1}$$

From this equation, two branches can be distinguished. The first is steady-state conduction, which is accomplished by assuming a null value for the left-hand side of the equation. The second is the transient-state conduction and is the following developed case.

Assuming constant thermophysical properties, equation (3.1) becomes as follows:

$$\rho\frac{\partial}{\partial t}\int u_i dV = -\int \vec{q}\cdot\vec{n}dS \tag{3.2}$$

This equation is applied to a certain domain discretized with uniform CVs and cell-centered nodes (*see Figure (3.1)*).

Let's focus on an internal control volume, denoted with the subscript P. A new mean internal energy is introduced to evaluate the whole energy inside the CV:

$$\overline{u_{iP}} = \frac{1}{V_P}\int u_i dV \tag{3.3}$$

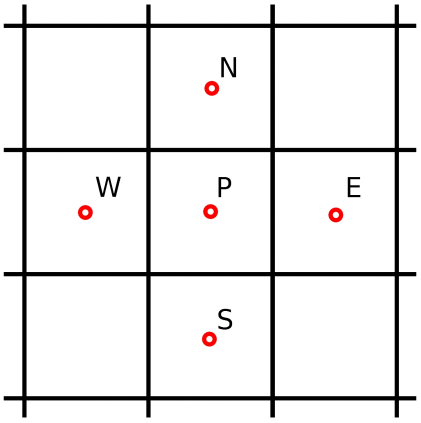


Figure 3.1: 2D spatial discretization

On the other side, it is assumed that the average energy can be written as a summation of the conduction heat flows entering and leaving the actual control volume. It is considered a two-dimensional domain with positive heat flow with positive coordinates axis. Subscripts stand for East, West, North, and South:

$$-\int \vec{q}\cdot\vec{n}dS = Q_E - Q_W + Q_N - Q_S = \sum Q \tag{3.4}$$

Taking equation (3.2) and introducing equations (3.3) and (3.4), the new equations is rewritten as follows:

$$\rho_P V_P \frac{\partial \overline{u_{iP}}}{\partial t} = \sum Q \tag{3.5}$$

Time integration is done at this point. Since time is also discretized in time steps, the integral is done between the current time and the next time step, t^n and t^{n+1} respectively. A new parameter β is introduced since the nature of the scheme can be implicit or explicit.

$$\int_{t^n}^{t^{n+1}} \rho_P V_P \frac{\partial \overline{u_{iP}}}{\partial t} dt = \int_{t^n}^{t^{n+1}} \sum Q dt \tag{3.6}$$

$$\rho_P V_P \frac{\overline{u_{iP}}^{n+1} - \overline{u_{iP}}^{n}}{\Delta t} = \beta \sum Q^{n+1} + (1+\beta)\sum Q^n \tag{3.7}$$

From now on, an implicit resolution is considered, being $\beta = 1$. In equation (3.7), the following relations can be introduced:

1. Internal energy definition:

$$\overline{u_{iP}} = c_P T_P \tag{3.8}$$

2. Fourier's law of Thermal Conduction with a second-order approximation

$$\dot{Q}_n = -\lambda \frac{\partial T}{\partial n} S_n = -\lambda \frac{T_X - T_P}{d_{XP}} S_x \tag{3.9}$$

being X a general node that can be East, West, North, or South.

Introducing the previous relations and expanding the summation term for the four possible locations mentioned before, the final equation results in the following:

$$\rho_P V_P c_P \frac{{T_P}^{n+1} - {T_P}^n}{\Delta t} = \left[\lambda_e \frac{T_E - T_P}{d_{PE}} S_e - \lambda_w \frac{T_P - T_W}{d_{PW}} S_w + \lambda_n \frac{T_N - T_P}{d_{PN}} S_n - \lambda_s \frac{T_P - T_S}{d_{PS}} S_s\right]^{n+1} \tag{3.10}$$

This equation can be easily simplified by introducing the discretization coefficients. These coefficients collect the different variables that are multiplying the node temperatures, and the equation results in a simpler form:

$$a_P T_P^{n+1} = a_W T_W^{n+1} + a_E T_E^{n+1} + a_S T_S^{n+1} + a_N T_N^{n+1} + b_P \tag{3.11}$$

It is important to remember that these coefficients can change between problems, depending on the conditions taken into consideration or the boundary conditions of each particular case. For this reason, a particular treatment of each coefficient will be done if it changes from this general form.

3.2 Code structure

The global algorithm to solve the problem is presented below:

1. Input data
   (a) Physical data, such as geometry, initial and physical properties, boundary conditions, etc.
   (b) Numerical data, such as convergence parameter, time step, mesh size, etc.
2. Previous calculations: vector definitions and mesh generation, assigning each CV its properties, using the *Block-off Method* depending on the material.
3. Initial temperature map: both real and estimated temperatures are initialized. The initial temperature value is assigned to the real temperature matrix at $t = 0$, T^n, and an arbitrary initial value is assigned to the estimated temperature matrix, T^n_{est}.
4. Time-convergence starting point:
   (a) Thermophysical properties calculations using the harmonic mean:

$$\lambda_{East} = \frac{d_{PE}}{\frac{d_{Pe}}{\lambda_P} + \frac{d_{eE}}{\lambda_E}} \tag{3.12}$$

   where d_{PE} stands for the distance between the actual and its east node, and d_{Pe} stands for the distance between the actual node and the east face of its CV. The operation is repeated for each face.

   (b) Discretization coefficients a_E, a_W, a_N, a_S, a_P and b_P. Particular treatment for each wall and the nodes inside. A detailed explanation is written in each particular case subsection.

(c) Temperature resolution: real temperature is solved by the previous coefficients and the estimated temperature T_{est}. The process is done node-by-node:

$$a_P T_P^{n+1} = a_W T_{W_{est}}^{n+1} + a_E T_{E_{est}}^{n+1} + a_S T_{S_{est}}^{n+1} + a_N T_{N_{est}}^{n+1} + b_P \tag{3.13}$$

(d) Convergence analysis: If the result converges, $max|T^{n+1} - T_{est}^{n+1}| < \delta$ the code goes to step 5. If not, the code updates the data, being the new estimated values the last computed ones $T_{est}^{n+1} = T^{n+1}$, and returns to step $4a$.

5. New time step: here two different criteria can be applied, depending on the steady-state achievement or a certain time achievement. This study in particular applies the second one, achieving up to 5000 seconds.

6. Final computations and results printing

3.3 Code verification

Code verification is an essential step as it helps to identify errors or bugs in the code. By verifying the code, it can be ensured that the code of the previous algorithm meets the intended specifications, is free of errors, and is reliable and efficient. For this reason, the following cases are solved using the same code and making the modifications needed.

The reason why these cases are chosen is that they have an analytical solution and taking it as a reference result, the code can be perfectly verified [6]. The problems will be solved as a one-dimensional steady-state problem, and the code is run until a steady state is reached.

At the end of the four verification cases, a particular problem is solved, which involves different boundary conditions and transient terms, as well as different materials. Thermophysical properties for all the verification cases will be taken from this last problem.

3.3.1 Case 1

The first case considers a single material with very high thermal conductivity, allowing simplification of the solution. The whole solid is in contact with an external fluid of constant and known properties α_g and T_g, and heat generation is not considered (*see Figure (3.2)*).

The analytical solution results from the following process:

$$\rho c_p \frac{\partial T}{\partial t} = \alpha_g (T - T_g) S_T \tag{3.14}$$

where S_T stands for the total surface of the solid in contact with the external fluid. Since the properties of the external fluid are constant over time, the previous equation has a direct solution:

$$T = T_g \tag{3.15}$$

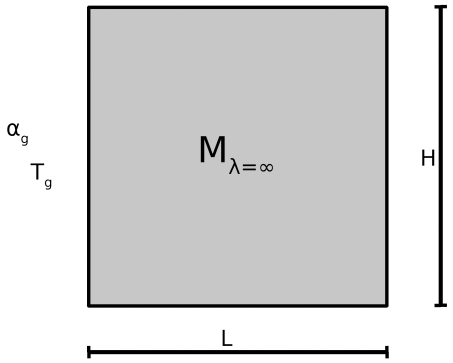


Figure 3.2: Verification case 1. Geometry and properties

As for the code, the following changes in the boundary conditions must be done:

1. **Thermal conductivity:** $\lambda = 10^8 \frac{W}{mK}$. Density and specific heat are also changed into the same value.
2. **Right, left, north, and south sides:** since the left side does not have west nodes, $a_W = 0$, and the surface of the north and south faces is null, so $a_N = a_S = 0$. Natural convection must be considered in a_P and b_P coefficients. The four sides follow the same scheme, so let's take the left side as an example:

$$\begin{aligned} a_E &= \frac{\lambda_e S_e}{d_{PE}} \\ a_W &= a_N = a_S = 0 \\ a_P &= a_E + \alpha_g S_e \\ b_P &= \alpha_g T_g S_e \end{aligned} \tag{3.16}$$

Considering the properties of the first material and the external fluid of the left side (*see Table (3.3)*), the results obtained are the following:

Table 3.1: 1^{st} verification case results. Comparison between analytical and numerical solutions

Analytical solution [K]	**Numerical solution [K]**
303	303.001

It can be observed that the relative error obtained is below 0.001%.

3.3.2 Case 2

The second case considers also a single material with the first material properties. The boundary conditions are adiabatic walls for the upper and lower ones and convection with external fluids A and B for the left and right walls. Also, heat generation is neglected.

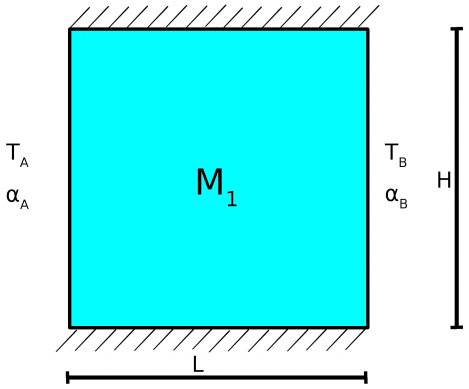


Figure 3.3: Verification case 2. Geometry and properties

The analytical solution results from the following process:

$$\rho c_p \frac{\partial T}{\partial t} = \lambda \left(\frac{\partial^2 T}{\partial x^2} \right) \tag{3.17}$$

Since the properties of the external fluid are constant over time, the previous equation can be integrated:

$$T(x) = C_1 x + C_2 \tag{3.18}$$

Applying boundary conditions on the left side, the second integration constant can be written in terms of the left wall temperature. Doing the same on the right side, the first integration constant can be written in terms of the right wall temperature:

$$T(x = 0) = T_{left} = C_2 \tag{3.19}$$

$$T(x = L) = T_{right} = C_1 L + T_{left} \tag{3.20}$$

To solve the two wall temperatures, an energy balance has to be applied, since the convection heat flux must be equal to the conduction heat flux in each wall.

$$\dot{q}_{conv} \mid_{(x=0)} = \dot{q}_{cond} \mid_{(x=0)} \tag{3.21}$$

$$\alpha_A (T_A - T_{left}) = -\lambda \frac{dT}{dx} \mid_{(x=0)} \tag{3.22}$$

$$\dot{q}_{cond} \mid_{(x=L)} = \dot{q}_{conv} \mid_{(x=L)} \tag{3.23}$$

$$-\lambda \frac{dT}{dx} \mid_{(x=L)} = \alpha_B (T_{right} - T_B) \tag{3.24}$$

The final analytical solution is the following one:

$$T(x) = \frac{-\alpha_A}{\lambda} (T_A - T_{left}) x + T_{left} \quad \text{where} \quad T_{left} = \frac{\alpha_A T_A \left(1 + \frac{\alpha_B L}{\lambda}\right) + \alpha_B T_B}{\alpha_B \left(1 + \frac{\alpha_A L}{\lambda}\right) + \alpha_A} \tag{3.25}$$

As for the code, the following changes in the boundary conditions must be done:

1. **Materials properties:** density, specific heat, and thermal conductivity are the same for the four materials.

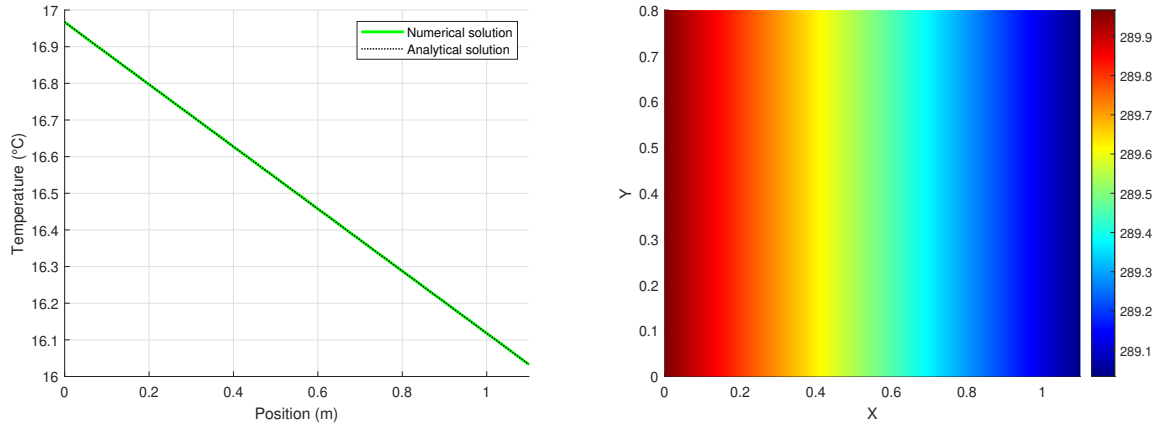
2. **Right and left sides:** both sides follow the same scheme. Take the left side as an example:

$$\begin{aligned} a_E &= \frac{\lambda_e S_e}{d_{PE}} \\ a_W &= a_N = a_S = 0 \\ a_P &= a_E + \alpha_g S_e \\ b_P &= \alpha_g T_g S_e \end{aligned} \tag{3.26}$$

3. **North and left sides:** this time, convection must not be considered in the upper and lower walls, since they are adiabatic. Both sides have the same adiabatic boundary condition, so let's take the north side as an example:

$$\begin{aligned} a_N &= 1 \\ a_S &= a_E = a_W = b_P = 0 \\ a_P &= 1 \end{aligned} \tag{3.27}$$

Considering the properties of the first material and the external fluid of the left side (*see Table (3.3)*), at $T_{left} = 33°C$ and $T_{right} = 0°C$, the results obtained are the following:



(a) Analytical and numerical solution comparison

(b) Numerical heat map

Figure 3.4: 2^{nd} verification case results. Comparison between analytical and numerical solutions

The maximum relative error obtained is below 0.003%.

3.3.3 Case 3

The third case considers two materials with the first and fourth material properties. The boundary conditions are adiabatic walls for the upper and lower ones and convection with external fluids A and B for the left and right walls. Again, heat generation is neglected.

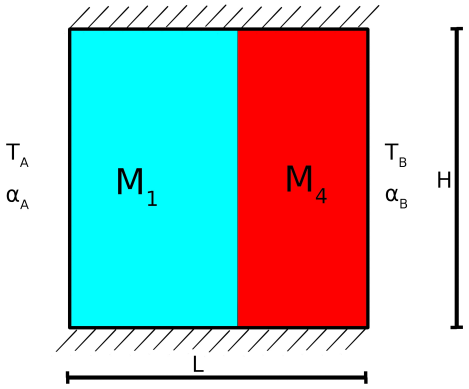


Figure 3.5: Verification case 3. Geometry and properties

The analytical solution results from the following process. This time, a different equation for each material must be found, and an additional energy balance between the two materials has to be done.

$$\rho c_p \frac{\partial T}{\partial t} = \lambda \left(\frac{\partial^2 T}{\partial x^2} \right) \tag{3.28}$$

Since the properties of the external fluid are constant over time, the previous equation can be integrated, and the two equations are found:

$$T_1(x) = C_1 x + C_2 \quad \text{and} \quad T_2(x) = C_3 x + C_4 \tag{3.29}$$

The solution is found by an alternative way to solve the previous C_n coefficients.

For the left wall, boundary condition and energy balance allow writing C_1 and C_2 in terms of the left wall temperature:

$$C_1 = \frac{-\alpha_A}{\lambda}(T_A - T_{left}) \quad \text{and} \quad C_2 = T_{left} \tag{3.30}$$

For the center wall, the contact between both materials, boundary conditions for the two materials, and an energy balance are applied:

$$T_{center} = C_1 e + C_2 \tag{3.31}$$

$$T_{center} = C_3 e + C_4 \tag{3.32}$$

$$-\lambda_1 C_1 = -\lambda_2 C_3 \tag{3.33}$$

Equations (3.31) and (3.32) must be equal, and equation (3.33) allows writing C_3 in terms of the left wall temperature.

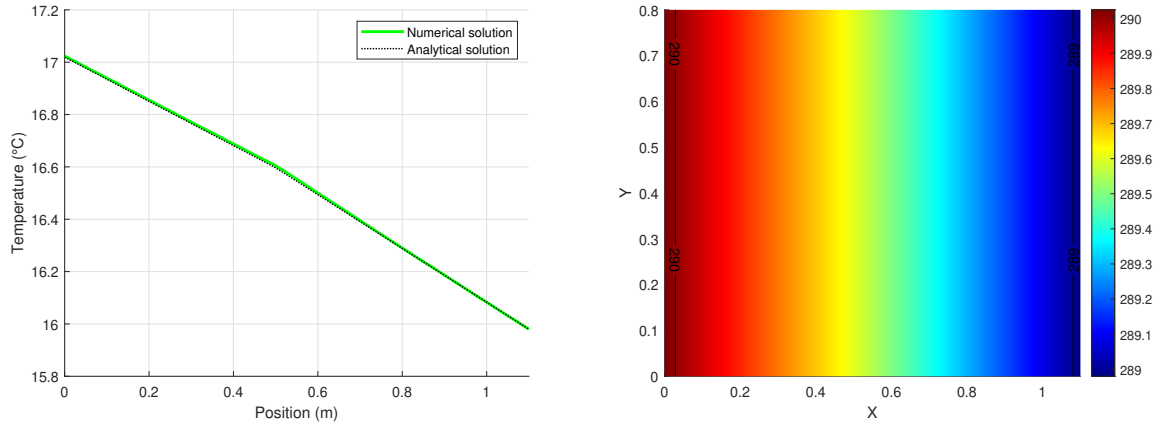
For the right wall, boundary condition and energy balance allow writing C_4 in terms of the right wall temperature, and this temperature can be also expressed in terms of the left wall.

Finally, the left wall can be expressed in terms of the external fluid temperatures. The final analytical solutions are found:

$$T_1(x) = \frac{-\alpha_A}{\lambda_1}(T_A - T_{left})x + T_{left} \quad \text{where} \quad T_{left} = \frac{T_B - T_A\alpha_A\left(\frac{-e}{\lambda_1} + \frac{e}{\lambda_2} - \frac{1}{\alpha_B} - \frac{l}{\lambda_2}\right)}{1 - \alpha_A\left(\frac{-e}{\lambda_1} + \frac{e}{\lambda_2} - \frac{1}{\alpha_B} - \frac{l}{\lambda_2}\right)} \tag{3.34}$$

$$T_2(x) = \frac{\alpha_A}{\lambda_2}(T_A - T_{left})(L - x) + T_{right} \quad \text{where} \quad T_{right} = \frac{\alpha_A}{\alpha_B}(T_A - T_{left}) + T_B \tag{3.35}$$

As for the code, the same changes as in the previous case must be done. Considering the properties of the first and fourth material and the external fluid of the left side (*see Table (3.3)*), at $T_{left} = 33°C$ and $T_{right} = 0°C$, the results obtained are the following:



(a) Analytical and numerical solution comparison

(b) Numerical heat map

Figure 3.6: 3^{rd} verification case results. Comparison between analytical and numerical solutions

3.3.4 Case 4

The fourth case is an extension of the previous one. Now, heat generation is considered in the first material.

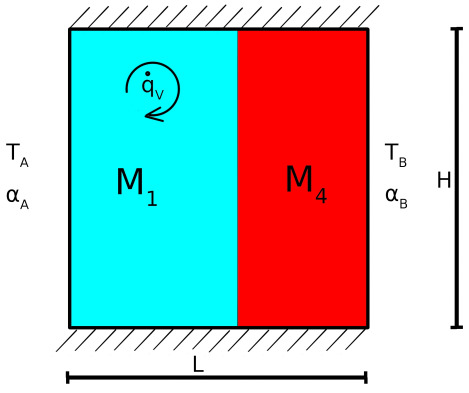


Figure 3.7: Verification case 4. Geometry and properties

The analytical solution also results from the previous process, adding a heat source to the equation of the first wall:

$$\rho c_p \frac{\partial T}{\partial t} = \lambda \left(\frac{\partial^2 T}{\partial x^2} \right) + \dot{q}_V \tag{3.36}$$

The integration results in the following two equations:

$$T_1(x) = \frac{-\dot{q}_V}{2\lambda_1} x^2 + C_1 x + C_2 \quad \text{and} \quad T_2(x) = C_3 x + C_4 \tag{3.37}$$

The solution is found following the same way as before to solve the previous C_n coefficients.

For the left wall, boundary condition and energy balance allow writing C_1 and C_2 in terms of the left wall temperature:

$$C_1 = \frac{-\alpha_A}{\lambda}(T_A - T_{left}) \quad \text{and} \quad C_2 = T_{left} \tag{3.38}$$

For the center wall, the contact between both materials, boundary conditions for the two materials, and an energy balance are applied. It must be seen that both temperature equations must be equal to the same center wall temperature:

$$T_1 center = \frac{-\dot{q}_V}{2\lambda_1} e^2 + C_1 e + C_2 \tag{3.39}$$

$$T_2 center = C_3 e + C_4 \tag{3.40}$$

$$-\lambda_1 \left(\frac{-\dot{q}_V}{\lambda_1} e + C_1 \right) = -\lambda_2 C_3 \tag{3.41}$$

Equation (3.41) allows writing C_3 in terms of the left wall temperature.

For the right wall, the boundary condition allows writing C_4 in terms of the right wall temperature, and the left wall through C_3.

$$C_4 = T_{right} - C_3 L \tag{3.42}$$

The energy balance allows writing the right wall temperature in terms of the left wall temperature:

$$T_{right} = \frac{\alpha_B T_B + \alpha_A (T_A - T_{left}) + \dot{q}_V e}{\alpha_B} \tag{3.43}$$

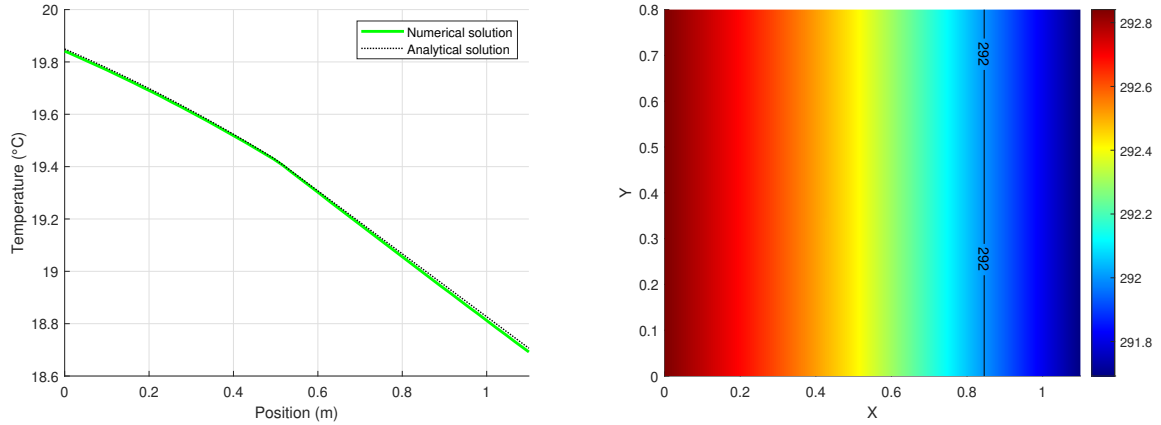
Finally, the values of the C_n coefficients are introduced in equations (3.39) and (3.40), together with the T_{right} equation to solve T_{left} in terms of the external fluid temperatures:

$$T_{left} = \frac{Tb + \alpha_A T_A \left(\frac{e}{\lambda_1} - \frac{e}{\lambda_2} + \frac{L}{\lambda_2} + \frac{1}{\alpha_B} \right) + \dot{q}_V e \left(\frac{e}{2\lambda_1} - \frac{e}{\lambda_2} + \frac{L}{\lambda_2} + \frac{1}{\alpha_B} \right)}{1 + \alpha_A \left(\frac{e}{\lambda_1} - \frac{e}{\lambda_2} + \frac{L}{\lambda_2} + \frac{1}{\alpha_B} \right)} \tag{3.44}$$

$$T_1(x) = \frac{-\dot{q}_V}{2\lambda_1} x^2 + \frac{-\alpha_A}{\lambda_1}(T_A - T_{left}) x + T_{left} \tag{3.45}$$

$$T_2(x) = \frac{-\alpha_A (T_A - T_{left}) - \dot{q}_V e}{\lambda_2} x + T_{right} + \frac{\alpha_A (T_A - T_{left}) L + \dot{q}_V e L}{\lambda_2} \tag{3.46}$$

As for the code, the same changes as in the previous cases must be done. Considering the properties of the first and fourth material and the external fluid of the left side (*see Table (3.3)*), at $T_{left} = 33°C$ and $T_{right} = 0°C$, and adding a heat generation of $\dot{q}_w = 100W/m^3$, the results obtained are the following:



(a) Analytical and numerical solution comparison

(b) Numerical heat map

Figure 3.8: 4^{th} verification case results. Comparison between analytical and numerical solutions

3.3.5 4-materials 3D rod

The main goal is the thermal analysis of a rod composed of four different materials through its section, with an ideal infinite length, different boundary conditions, and transient state once the code has been successfully verified. The geometry is the following:

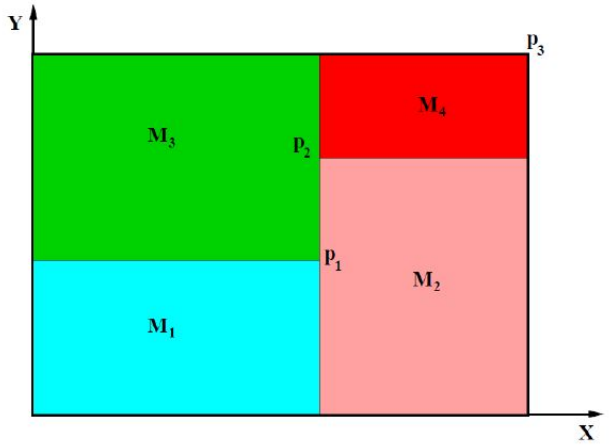


Figure 3.9: Materials distribution. Extracted from [7]

The following table sums up the physical properties of the different materials and the boundary conditions for each wall of the rod, respectively. The following properties are the ones used for the previous verification cases:

Table 3.2: Thermophysical properties of the materials

	ρ **(Kg/m^3)**	c_P **(J/KgK)**	λ **(W/mK)**
M1	1500	750	170
M2	1600	770	140
M3	1900	810	200
M4	2500	930	140

Table 3.3: Specific boundary conditions

Wall	**Boundary condition**
Bottom	Isothermal wall at $T_b = 23°C$
Top	Uniform incident heat flux $\dot{q}_w = 54.55W/m^2$ (negative on Y)
Left	Convection with external fluid at $T_l = 33°C$ and $\alpha_l = 9W/m^2K$
Right	Uniform transitory wall at $T_r(t) = 8 + 0.005t$

As a last condition, it is established that the initial rod temperature is $8°C$.

As said before, the discretization coefficients vary depending on the wall. For this reason, a particular treatment must be done for each side of the rod:

1. **Left side:** since the nodes of this side don't have west nodes, is imposed that $a_W = 0$, and the surface of the north and south faces is null, so $a_N = a_S = 0$. Convection with the east side must be applied for a_E. The rest of the coefficients are defined as follows:

$$\begin{aligned} a_E &= \frac{\lambda_e S_e}{d_{PE}} \\ a_P &= a_E + \frac{\rho_P c_P V_P}{\Delta_t} + \alpha_l S_w \\ b_P &= \frac{\rho_P c_P V_P}{\Delta_t} T_P^n + \alpha_l T_l S_w \end{aligned} \tag{3.47}$$

2. **Right side:** since the nodes of this side don't have east nodes, is imposed that $a_E = 0$, and the surface of the north and south faces is null, so $a_N = a_S = 0$. In addition, this side corresponds to a uniform transitory wall, which does not depend on its west nodes, so $a_W = 0$, and only b_P takes a non-null value:

$$\begin{aligned} a_P &= 1 \\ b_P &= (8 + 0.005t) + 273 \end{aligned} \tag{3.48}$$

3. **Top side:** since the nodes of this side don't have north nodes, is imposed that $a_N = 0$, and the surface of the east and west faces is null, so $a_E = a_W = 0$. Conduction with the north side must be applied.

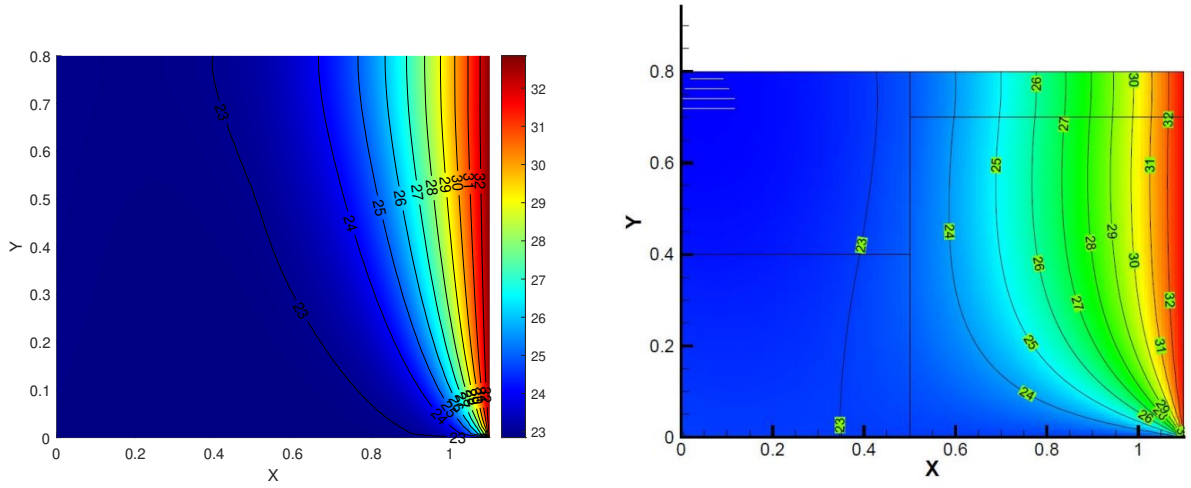
The rest of the coefficients are defined as follows:

$$
\begin{aligned}
a_S &= \frac{\lambda_w S_w}{d_{PW}} \\
a_P &= a_S + \frac{\rho_P c_P V_P}{\Delta_t} \\
b_P &= \frac{\rho_P c_P V_P}{\Delta_t} T_P^n + \dot{q_w} S_w
\end{aligned} \tag{3.49}
$$

4. **Bottom side:** since this side's nodes are at a constant temperature, the a_P coefficient only relies on this temperature, which is assigned by the b_P coefficient. The rest of the coefficients take a null value $a_E = a_W = a_N = a_S = 0$:

$$
\begin{aligned}
a_P &= 1 \\
b_P &= T_b
\end{aligned} \tag{3.50}
$$

The temperature map obtained after 5000 seconds is the one below, together with the reference solutions:



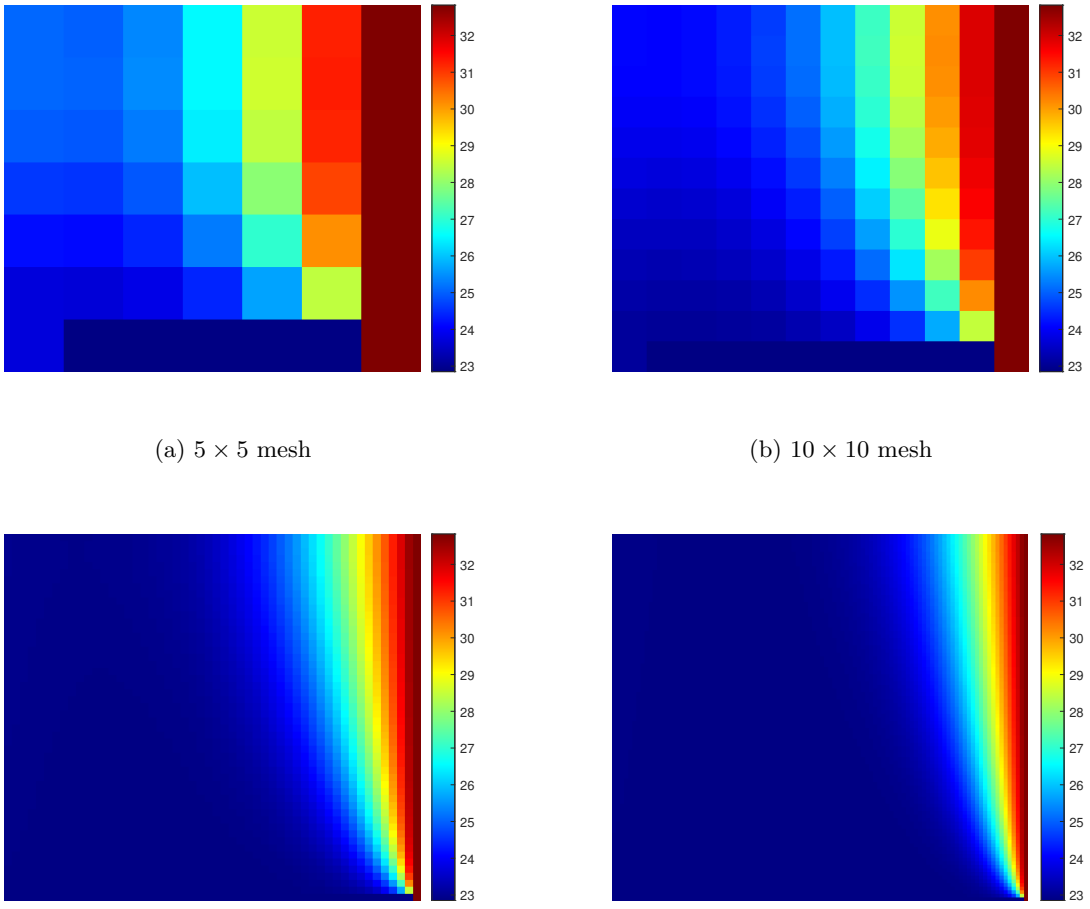
(a) Numerical solutions (b) Reference solutions. Extracted from [7]

Figure 3.10: Temperature map after 5000 seconds and 50×50 mesh

3.3.6 Mesh size influence

The mesh influence is important as it can prevent the code from reaching convergence. Also, the accuracy of the results depends on this parameter.

As can be seen below, four different results are obtained with four different uniform meshes (*see Figure (3.11)*). The coarse mesh of the two first cases results in a significant inaccuracy compared to the reference result, provided by CTTC. The results obtained with a finer mesh seem to converge in the same result as the reference one.



(a) 5×5 mesh

(b) 10×10 mesh

(c) 50×50 mesh

(d) 100×100 mesh

Figure 3.11: Temperature map after 5000 seconds for different meshes

3.4 Discussion

In conclusion, this chapter presented the verification of the heat conduction code through the analysis of four different verification cases and a benchmark problem. Each verification case allowed the comparison of the numerical results with analytical solutions or reference data.

The first four verification cases involved one-dimensional steady-state heat conduction problems with different boundary conditions and internal configurations. The code accurately reproduced the analytical solution, confirming its capability to handle simple heat transfer problems. Finally, the benchmark problem of the four materials, which is widely used in the heat conduction field, also demonstrated the excellent agreement between the numerical results and the benchmark data further reinforcing the reliability and validity of the numerical approach.

These results provide confidence in the code's ability to handle a wide range of heat transfer scenarios, laying a solid foundation for its application to extension cases in subsequent chapters.

Chapter 4

Convection and Diffusion

This fourth chapter is focused on the mathematical study and the appliance of the general Convection-Diffusion equation For this purpose, a previous mathematical study is done, and four benchmark problems are solved with the developed algorithm.

4.1 The Convection-Diffusion Equation

The convection-diffusion equation comes from summarizing all the governing equations of the heat transfer by convection, also known as the Navier-Stokes equations [8]. Composed of mass conservation (4.1), momentum conservation (4.2), and energy conservation (4.3), the Navier-Stokes equations are expressed in differential form as follows:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \tag{4.1}$$

$$\frac{\partial (\rho \vec{v})}{\partial t} + \nabla \cdot (\rho \vec{v} \vec{v}) = \nabla \cdot (\mu \nabla \vec{v}) + \{\nabla \cdot (\vec{\tau} - \mu \nabla \vec{v}) - \nabla p + \rho \vec{g}\} \tag{4.2}$$

$$\frac{\partial (\rho T)}{\partial t} + \nabla \cdot (\rho \vec{v} T) = \nabla \cdot \left(\frac{\lambda}{c_v} \nabla T \right) + \left\{ \frac{-\nabla \cdot \vec{q}^R - p \nabla \cdot \vec{v} + \vec{\tau} : \nabla \vec{v}}{c_v} \right\} \tag{4.3}$$

These previous equations can be expressed in a single one, named the general Convection-Diffusion Equation:

$$\frac{\partial (\rho \phi)}{\partial t} + \nabla \cdot (\rho \vec{v} \phi) = \nabla \cdot (\Gamma \nabla \phi) + S_\phi \tag{4.4}$$

Being ϕ a general variable, Γ a diffusion coefficient, and S_ϕ the source term, the following table gives each specific value of the previous variables to reproduce the governing equations:

Table 4.1: Specific values of ϕ, Γ, and S_ϕ

Governing equation	ϕ	Γ	S_ϕ
Mass conservation	1	0	0
Momentum conservation	$\vec{v}$	μ	$\nabla \cdot (\vec{\tau} - \mu \nabla \vec{v}) - \nabla p + \rho \vec{g}$
Energy conservation	T	$\frac{\lambda}{c_v}$	$\frac{-\nabla \cdot \vec{q}^R - p \nabla \cdot \vec{v} + \vec{\tau} : \nabla \vec{v}}{c_v}$

For numerical discretization, FVM is applied, with non-overlapping CVs and cell-centered nodes. An implicit scheme is considered for the unsteady terms. With all these considerations taken into account, the general equation (4.4) is integrated and results as follows:

$$\begin{aligned} &\frac{\rho_P \phi_P - \rho_P^0 \phi_P^0}{\Delta t} V_P + \dot{m}_e(\phi_e - \phi_P) - \dot{m}_w(\phi_P - \phi_w) + \dot{m}_n(\phi_n - \phi_P) - \dot{m}_s(\phi_P - \phi_s) \\ &= D_e(\phi_E - \phi_P) - D_w(\phi_P - \phi_W) + D_n(\phi_N - \phi_P) - D_s(\phi_P - \phi_S) \end{aligned} \tag{4.5}$$

Note that each term of the equation is referred to the next time step, except the ones expressed with superscript x^0, which are referred to the previous time step.

Also, note that the convection terms are expressed as a function of the ϕ value at the CV face (ϕ_e), but the diffusion terms are referred to the ϕ value at the adjacent node (ϕ_E). To avoid this situation and standardize the nomenclature, different schemes are presented.

4.1.1 Central-Difference scheme - CDS

Being the simplest scheme assumes a linear ϕ distribution between adjacent nodes. With this assumption, the following relation is applied:

$$\phi_e - \phi_P = f_e(\phi_E - \phi_P) \tag{4.6}$$

where f_e stands for the ratio of the distance between the node and the face, and the distance between both nodes.

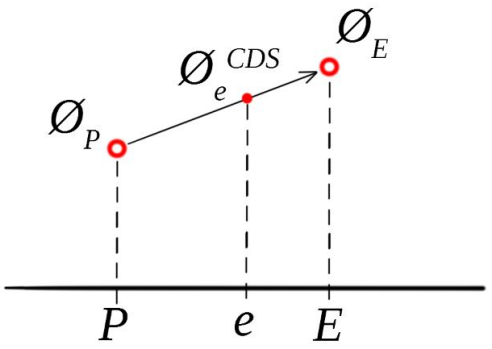


Figure 4.1: CDS scheme

Introducing this relation into equation (4.5) and grouping the different terms referring to the nodes, the equation is discretized as follows:

$$a_P\phi_P = a_E\phi_E + a_W\phi_W + a_N\phi_N + a_S\phi_S + b_P \tag{4.7}$$

where

$$a_E = D_e - \dot{m}_e f_e \qquad a_W = D_w - \dot{m}_w f_w \qquad a_N = D_n - \dot{m}_n f_n \qquad a_S = D_s - \dot{m}_s f_s \tag{4.8}$$

$$a_P = a_E + a_W + a_N + a_S + \frac{\rho_P^0 V_P}{\Delta t} \tag{4.9}$$

$$b_P = \frac{\rho_P^0 V_P}{\Delta t}\phi_P^0 \tag{4.10}$$

$$D_e = \frac{\Gamma_e S_e}{d_{PE}} \qquad D_w = \frac{\Gamma_w S_w}{d_{PW}} \qquad D_n = \frac{\Gamma_n S_n}{d_{PN}} \qquad D_s = \frac{\Gamma_s S_s}{d_{PS}} \tag{4.11}$$

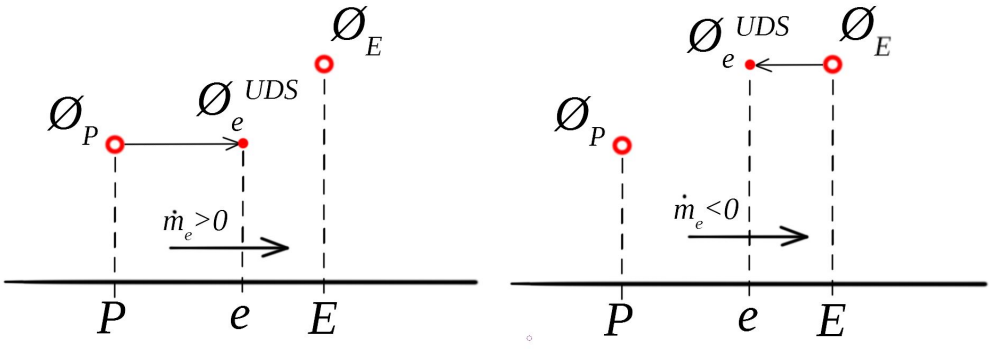
$$f_e = \frac{d_{Pe}}{d_{PE}} \qquad f_w = \frac{d_{Pw}}{d_{PW}} \qquad f_n = \frac{d_{Pn}}{d_{PN}} \qquad f_s = \frac{d_{Ps}}{d_{PS}} \tag{4.12}$$

Being a 2nd-order accurate scheme, CDS presents stability problems.

4.1.2 Upwind-Difference scheme - UDS

The UDS is defined by the assumption that for incompressible flows, or gases at low Mach number, convective terms are more influenced by upstream nodes than downstream conditions. For this reason, the following relation is applied:

$$\dot{m}_e(\phi_e - \phi_P) = \frac{\dot{m}_e - |\dot{m}_e|}{2}(\phi_E - \phi_P) \tag{4.13}$$



(a) Positive UDS scheme (b) Negative UDS scheme

Figure 4.2: UDS scheme

Introducing this relation into equation (4.5) and grouping the different terms referring to the nodes, the same

equation is obtained (4.7). This time, the coefficients are the following:

$$a_E = D_e - \frac{\dot{m}_e - |\dot{m}_e|}{2} \qquad a_W = D_w - \frac{\dot{m}_w - |\dot{m}_w|}{2} \tag{4.14}$$

$$a_N = D_n - \frac{\dot{m}_n - |\dot{m}_n|}{2} \qquad a_S = D_s - \frac{\dot{m}_s - |\dot{m}_s|}{2}$$

$$a_P = a_E + a_W + a_N + a_S + \frac{\rho_P^0 V_P}{\Delta t} \tag{4.15}$$

$$b_P = \frac{\rho_P^0 V_P}{\Delta t}\phi_P^0 \tag{4.16}$$

$$D_e = \frac{\Gamma_e S_e}{d_{PE}} \qquad D_w = \frac{\Gamma_w S_w}{d_{PW}} \qquad D_n = \frac{\Gamma_n S_n}{d_{PN}} \qquad D_s = \frac{\Gamma_s S_s}{d_{PS}} \tag{4.17}$$

UDS is much more stable than the previous scheme but is 1st-order accurate. Since a 2nd-order accurate and stable scheme is searched, the next scheme is presented.

4.1.3 Exponential-Difference scheme - EDS

This scheme assumes a ϕ distribution based on a simplified form of the general Convection-Diffusion equation without source terms. The same relation as for the CDS scheme is applied (4.6), and the equation is discretized as follows:

$$a_P\phi_P = a_E\phi_E + a_W\phi_W + a_N\phi_N + a_S\phi_S + b_P \tag{4.18}$$

where

$$a_E = D_e - \dot{m}_e f_e \qquad a_W = D_w - \dot{m}_w f_w \qquad a_N = D_n - \dot{m}_n f_n \qquad a_S = D_s - \dot{m}_s f_s \tag{4.19}$$

$$a_P = a_E + a_W + a_N + a_S + \frac{\rho_P^0 V_P}{\Delta t} \tag{4.20}$$

$$b_P = \frac{\rho_P^0 V_P}{\Delta t}\phi_P^0 \tag{4.21}$$

$$D_e = \frac{\Gamma_e S_e}{d_{PE}} \qquad D_w = \frac{\Gamma_w S_w}{d_{PW}} \qquad D_n = \frac{\Gamma_n S_n}{d_{PN}} \qquad D_s = \frac{\Gamma_s S_s}{d_{PS}} \tag{4.22}$$

$$f_e = \frac{e^{P_e \frac{d_{Pe}}{d_{PE}}} - 1}{e^{P_e}} \qquad f_w = \frac{e^{P_w \frac{d_{Pw}}{d_{PW}}} - 1}{e^{P_w}} \qquad f_n = \frac{e^{P_n \frac{d_{Pn}}{d_{PN}}} - 1}{e^{P_n}} \qquad f_s = \frac{e^{P_s \frac{d_{Ps}}{d_{PS}}} - 1}{e^{P_s}} \tag{4.23}$$

where P_x stands for the local Péclet number, and it is computed as follows:

$$P_e = \frac{\rho_e v_e d_{PE}}{\Gamma_e} \qquad P_w = \frac{\rho_w v_w d_{PW}}{\Gamma_w} \qquad P_n = \frac{\rho_n v_n d_{PN}}{\Gamma_n} \qquad P_s = \frac{\rho_s v_s d_{PS}}{\Gamma_s} \tag{4.24}$$

EDS is also stable and it is 2nd-order accurate, but when it comes to a large Péclet number, the solution tends to be the same as for UDS. For this reason, high-resolution schemes (HRS) such as SUDS or QUICK are applied.

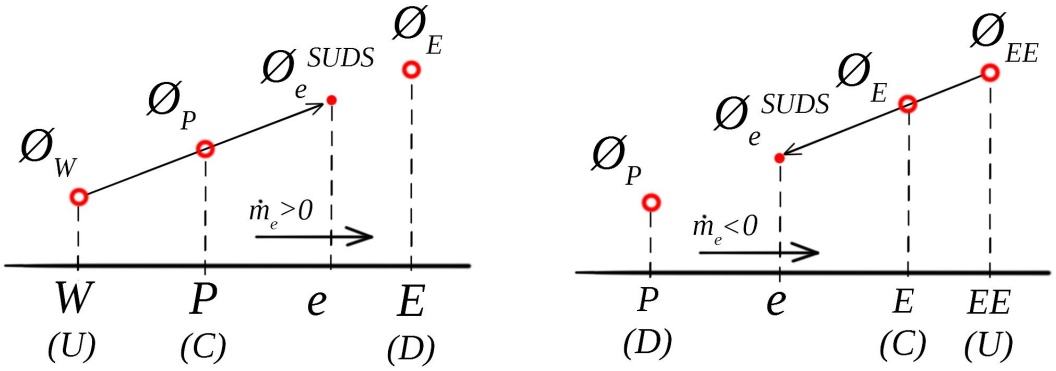
4.1.4 Second-order Upwind linear extrapolation - SUDS

This first HRS assumes a linear extrapolation but with the two upstream nodes. Located at the face position, the nomenclature for the three nodes is the following:

D = downstream node

C = first upstream node

U = most upstream node



(a) Positive SUDS scheme

(b) Negative SUDS scheme

Figure 4.3: SUDS scheme

The relation for solving ϕ at the desired face is a little trickier than the schemes mentioned before since the ϕ result is obtained through its dimensionless form as it follows:

$$\hat{\phi}_f = \frac{\hat{x}_f}{\hat{x}_C}\hat{\phi}_C \tag{4.25}$$

where $\hat{\phi}_f$ stands for the located face dimensionless ϕ value, and:

$$\hat{\phi} = \frac{\phi - \phi_U}{\phi_D - \phi_U} \tag{4.26}$$

and

$$\hat{x} = \frac{x - x_U}{x_D - x_U} \tag{4.27}$$

To better understand the whole process, the following simple algorithm is applied to solve ϕ_e:

1. Input geometrical and ϕ^{est} data, and mass fluxes
2. Dimensional values assignation: x_D, x_C, and x_U, and ϕ_D, ϕ_C, and ϕ_U
3. Dimensionless values evaluation: $\hat{\phi}_C$ with equation (4.26), and $\hat{x}_C$ together with $\hat{x}_e$, both with equation (4.27)
4. $\hat{\phi}_e$ evaluation with equation (4.25)
5. ϕ_e evaluation, rearranging equation (4.26)

The following relation is applied to solve the next time step, as it has been done in the previous schemes:

$$\phi_f^{HRS} - \phi_P = (\phi_f^{UDS} - \phi_P) + (\phi_f^{HRS^{prev}} - \phi_f^{UDS^{prev}}) \tag{4.28}$$

where $\phi_f^{HRS^{prev}}$ and $\phi_f^{UDS^{prev}}$ are calculated with the values obtained in the previous iteration. The discretized equation takes the following form:

$$a_P\phi_P = a_E\phi_E + a_W\phi_W + a_N\phi_N + a_S\phi_S + b_P \tag{4.29}$$

where

$$a_E = D_e - \frac{\dot{m}_e - |\dot{m}_e|}{2} - \dot{m}_e(\phi_e^{HRS^{prev}} - \phi_e^{UDS^{prev}}) \qquad a_W = D_w - \frac{\dot{m}_w - |\dot{m}_w|}{2} + \dot{m}_w(\phi_w^{HRS^{prev}} - \phi_w^{UDS^{prev}}) \tag{4.30}$$

$$a_N = D_n - \frac{\dot{m}_n - |\dot{m}_n|}{2} - \dot{m}_n(\phi_n^{HRS^{prev}} - \phi_n^{UDS^{prev}}) \qquad a_S = D_s - \frac{\dot{m}_s - |\dot{m}_s|}{2} + \dot{m}_s(\phi_s^{HRS^{prev}} - \phi_s^{UDS^{prev}}) \tag{4.31}$$

$$a_P = a_E + a_W + a_N + a_S + \frac{\rho_P^0 V_P}{\Delta t} \tag{4.32}$$

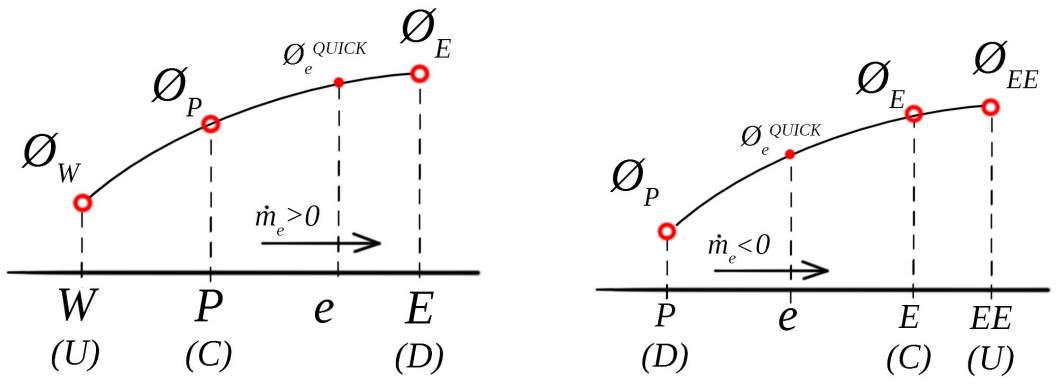
$$b_P = \frac{\rho_P^0 V_P}{\Delta t}\phi_P^0 \tag{4.33}$$

$$D_e = \frac{\Gamma_e S_e}{d_{PE}} \qquad D_w = \frac{\Gamma_w S_w}{d_{PW}} \qquad D_n = \frac{\Gamma_n S_n}{d_{PN}} \qquad D_s = \frac{\Gamma_s S_s}{d_{PS}} \tag{4.34}$$

4.1.5 Quadratic Interpolation for Convective Kinematics - QUICK

This second HRS assumes a quadratic extrapolation also with the two upstream nodes, and it could be up to the third-order if desired. The process followed to solve the dimensional ϕ_e value is exactly the same as before, but the relation for the dimensionless located face ϕ value is changed for the following one:

$$\hat{\phi}_f = \hat{x}_f + \frac{\hat{x}_f(\hat{x}_f - 1)}{\hat{x}_C(\hat{x}_C - 1)}(\hat{\phi}_C - \hat{x}_C) \tag{4.35}$$



(a) Positive QUICK scheme

(b) Negative QUICK scheme

Figure 4.4: QUICK scheme

4.2 Code structure

The following lines sum up the general algorithm used to solve the ϕ distribution for every case. The same code can be used to solve the four different cases by changing boundary conditions and using the previous schemes by changing the expressions to evaluate convection and diffusion terms.

1. Input data:
   (a) Physical data, such as geometry, initial values, physical properties, and boundary conditions.
   (b) Numerical data, such as convergence parameter, time step, mesh size, etc.
2. Previous calculations: vector definitions and mesh generation.
3. Initial maps: $\phi^0 = \phi(t = 0, x, y)$
4. Time-convergence starting point: $t = t + \Delta t$
   (a) Initial estimated values: $\phi^{est} = \phi^0$
   (b) Convection and diffusion terms: $\dot{m}, D_x, f_x$
   (c) Discretization coefficients evaluation: $a_E, a_W, a_N, a_S, a_P, b_P$
   (d) ϕ resolution:
   $$a_P\phi_{new} = a_E\phi_E + a_W\phi_W + a_N\phi_N + a_S\phi_S + b_P \tag{4.36}$$
   (e) Convergence analysis: if the result converges, $|max(\phi_{new} - \phi_{est})| < \delta$ the code goes to step 5. If not, the code updates data, $\phi_{est} = \phi_{new}$, and goes back to step $4a$.
5. Time convergence analysis: $t \geqslant t_{final}$
6. Data update: $\phi_0 = \phi_{new}$
7. Final computations $\phi_{analytical}$, and results printing

4.3 Code verification

The code verification is done by four different benchmark problems. Each problem has the same main characteristics, listed below:

1. Steady-state convection-diffusion cases.
2. Velocity field is known.
3. 2D cases, with uniform mesh and cell-centered nodes.
4. Density and diffusion coefficients are known constant values. Source terms take null value for every case.

Particular treatment of the boundary conditions is done for each case. The following sections present these boundary conditions and the obtained results for each case, along with the reference results and, when possible, analytic solution.

4.3.1 Parallel flow

As said before, is a steady-state convection-diffusion case where ϕ has a one-dimensional variation in the same direction as the flow in a $L \times L$ domain. The flow follows the equation below:

$$\vec{v}(x, y) = (v_0, 0) \tag{4.37}$$

The following boundary conditions are applied:

1. **Left side:** the Dirichlet boundary condition is applied, which establishes that all these nodes have a constant $\phi = \phi_{in}$ value.
2. **Right side:** the same Dirichlet boundary condition is applied, which also establishes that all these nodes have a constant $\phi = \phi_{out}$ value.
3. **Upper and lower sides:** the Neumann boundary condition is applied, which establishes the following relation:

$$\frac{\partial \phi}{\partial y} = 0 \tag{4.38}$$

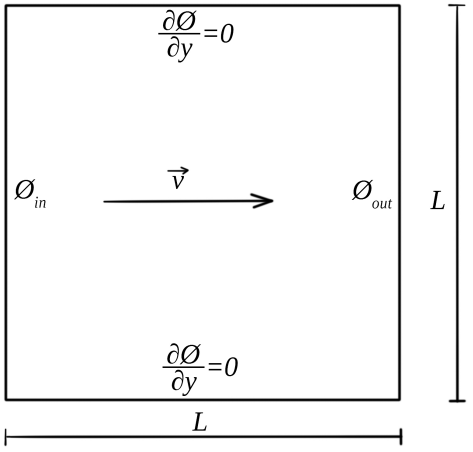
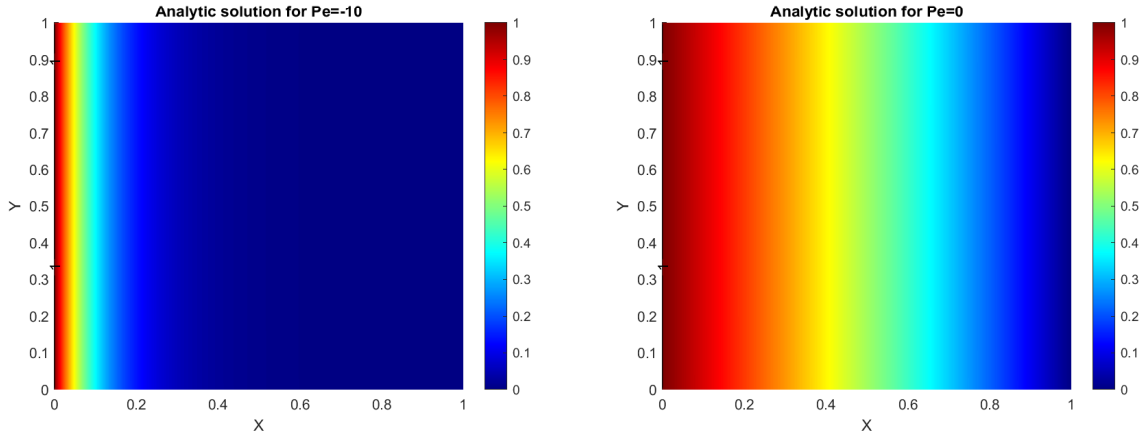


Figure 4.5: Parallel flow scheme. Geometry and boundary conditions

This particular case has an analytic solution, described by the following equation [9]:

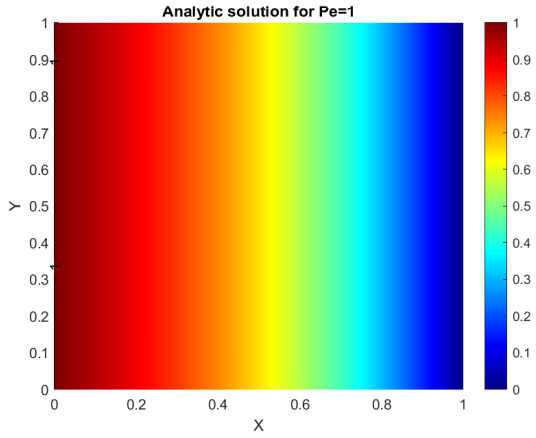
$$\frac{\phi - \phi_{in}}{\phi_{out} - \phi_{in}} = \frac{e^{\frac{x}{L} Pe} - 1}{e^{Pe} - 1} \tag{4.39}$$

As can be seen, the solution depends on the Péclet number of the problem. The obtained results for different Péclet numbers are presented in the figure below:

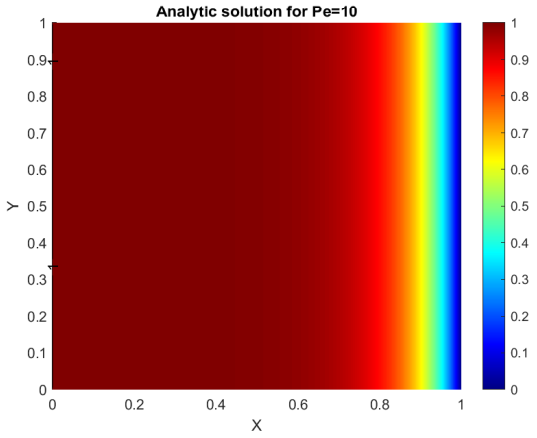


(a) Pe= -10

(b) Pe=0



(c) Pe=1



(d) Pe=10

Figure 4.6: Analytic solution for different Péclet numbers and 100×100 mesh

Some significant conclusions can be extracted from the above-mentioned figure. For $Pe = 0$, (*see Figure (4.6b)*), the case becomes a full conduction problem since the diffusion coefficient takes an infinite value and the domain act as a solid. It is easy to see that the behavior of the ϕ is very similar to the one obtained in the previous chapter, the second verification case. As the Péclet number increase or decreases, the diffusion term becomes larger, and, thus, the conduction term is no longer the dominant term of the problem. Some tests with $Pe << 1$ and $Pe >> 1$ have been done in order to see this growth of the diffusion influence (*see Figures (4.6a) and (4.6d)*).

Since this analytic solution comes from the *EDS*, a comparison between analytical and numerical results is done for the same Péclet numbers. As can be seen, the numerical results are pretty accurate with the analytical ones:

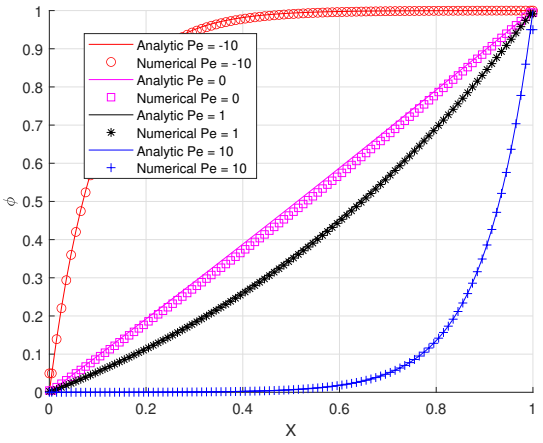


Figure 4.7: Comparison between analytical and numerical results for EDS

4.3.2 Vertical flow

This case is a modification of the previous one. Now the flow is vertical, as the equation shows:

$$\vec{v}(x, y) = (0, v_0) \tag{4.40}$$

The exact same boundary conditions as before are applied, where the left and right sides act as inlet and outlet, and the upper and lower sides apply the Neumann condition (4.38).

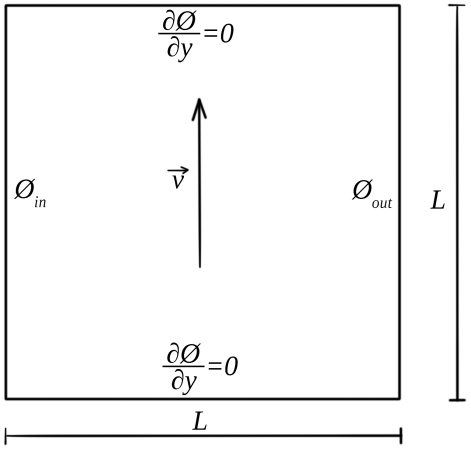


Figure 4.8: Vertical flow scheme. Geometry and boundary conditions

This new case has another analytic solution, described by the following equation:

$$\phi = \phi_{in} + \frac{\phi_{out} - \phi_{in}}{L} x \tag{4.41}$$

As can be seen, the solution does not depend on the Péclet number, and the results are expected to be the same as the ones obtained for $Pe = 0$ in the previous case. The obtained results are the following:

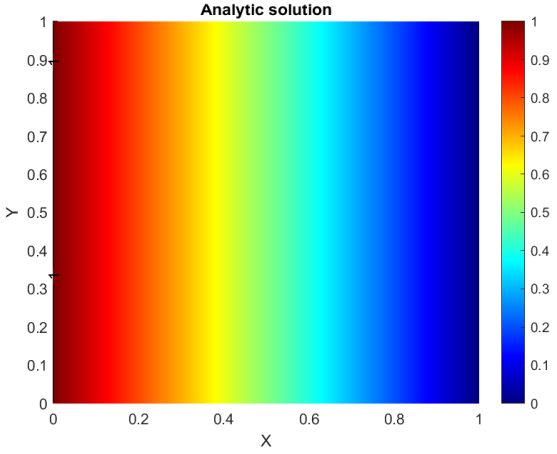


Figure 4.9: Analytic solution with 100×100 mesh

As it was expected, the results obtained results with the different schemes are very similar to the analytical ones. In order to better see the difference between each result, the absolute error compared with the analytic solution is plotted below:

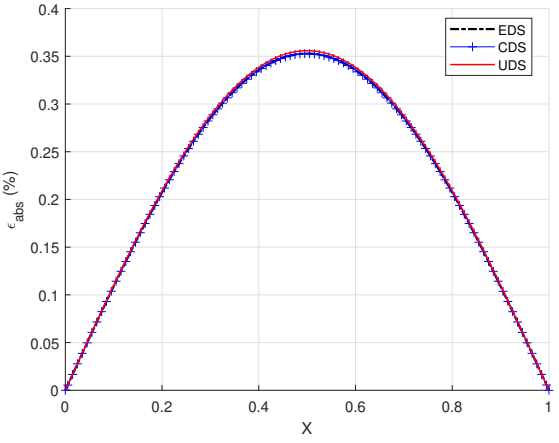


Figure 4.10: Comparison between analytical and numerical results for EDS, UDS and CDS

Only a 0.35% of error is observed with UDS, while CDS and EDS provide the exact same values along the domain, and are slightly lower than the ones provided by the UDS.

4.3.3 Diagonal flow

The third case has a diagonal flow, as the following equation shows:

$$\vec{v}(x, y) = (v_0 cos(\alpha), v_0 sin(\alpha)) \tag{4.42}$$

where $\alpha = 45°$. The following boundary conditions are applied:

1. **Left and upper sides:** the Dirichlet boundary condition is applied, which establishes that all these nodes have a constant $\phi = \phi_{low}$ value.

2. **Right and lower sides:** the same Dirichlet boundary condition is applied, which also establishes that all these nodes have a constant $\phi = \phi_{high}$ value.

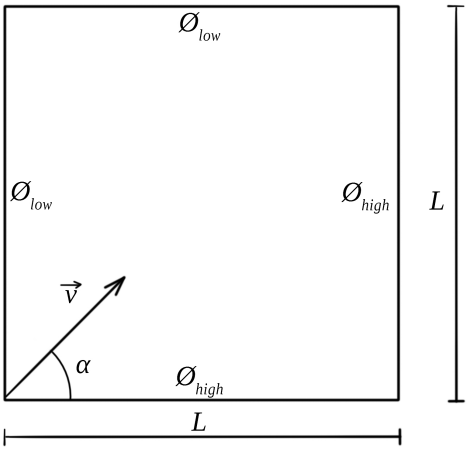


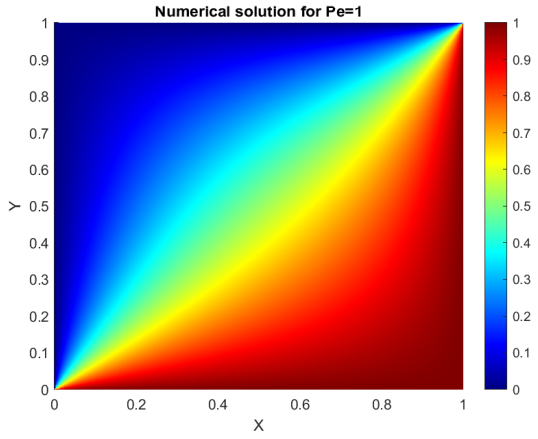
Figure 4.11: Diagonal flow scheme. Geometry and boundary conditions

This case is expected to be simulated for $Pe = \infty$. For this reason, UDS will only be used since EDS and CDS present stability problems for high Péclet numbers. The expected results are the following ones:

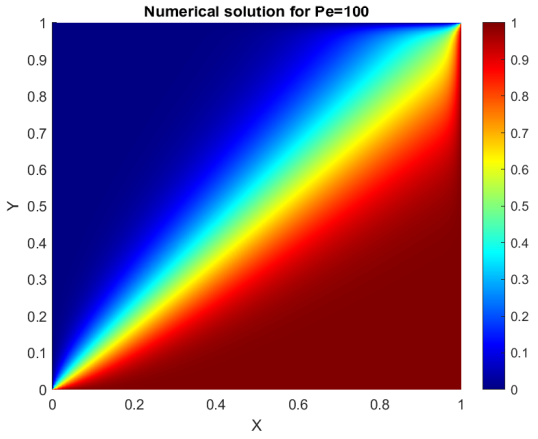
1. $\phi = \phi_{low}$ above the diagonal.
2. $\phi = \phi_{high}$ below the diagonal.

The following figure presents the obtained results for different Péclet numbers. For higher Péclet values, the diagonal flow starts to stretch, reducing the transition area from the lowest to the highest value, which means that the gradient of values between consecutive nodes increases rapidly.

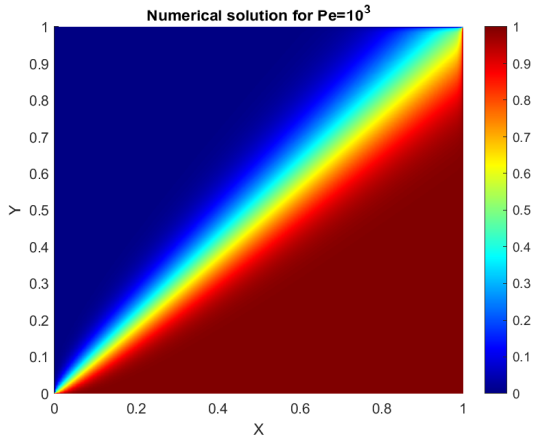
As can easily be observed, no change in the results is appreciated between the last two Péclet number simulations. This is due to the false diffusion. A coarse mesh is being used, and thus, bigger errors are computed since the distances between nodes are not small enough to capture the smallest variations. For this reason, the results for $Pe = \infty$ ($Pe = 10^7$) are recomputed using finer meshes. This way, the above-mentioned gradients of values can be correctly computed. The new results are presented in Figure (4.13), where the change can now be appreciated.



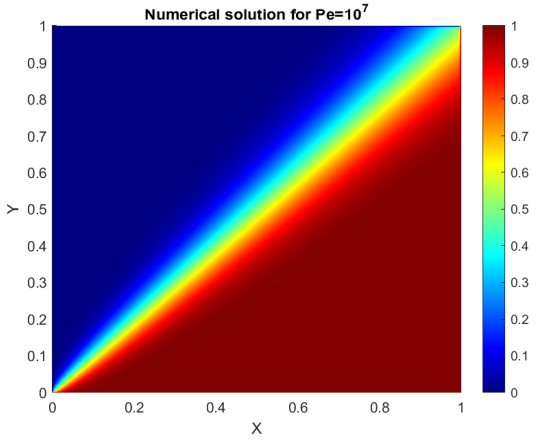
(a) Pe=1



(b) Pe=100

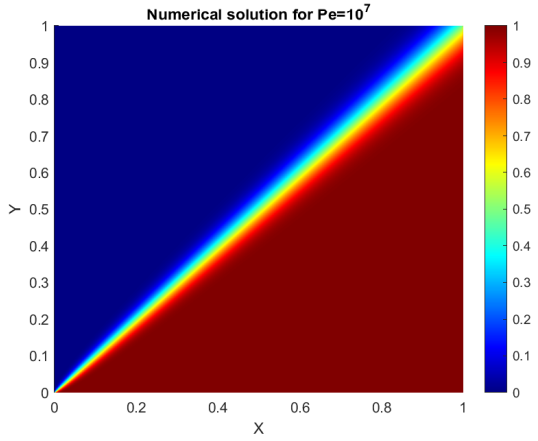


(c) Pe=1000

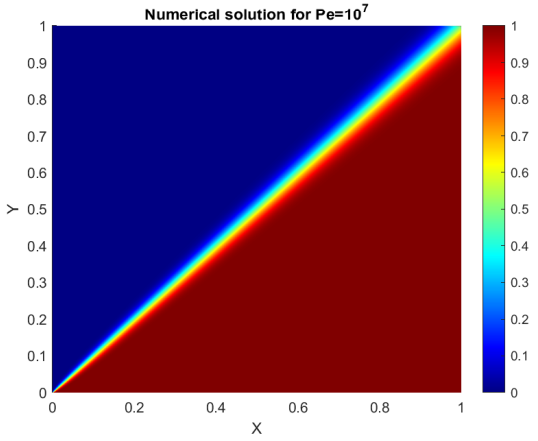


(d) Pe= 10^7

Figure 4.12: Numerical solution for different Péclet numbers and 100×100 mesh with UDS



(a) Pe= 10^7 and 500×500 mesh



(b) Pe= 10^7 and 1000×1000 mesh

Figure 4.13: Numerical solution for $Pe = \infty$ ($Pe = 10^7$) with finer meshes and UDS

4.3.4 Smith-Hutton problem

This case has a particular treatment, as it becomes a rectangular domain, $2L \times L$. Regarding the flow, the velocity map is defined as the following equation shows:

$$\vec{v}(x, y) = (2y(1 - x^2), -2x(1 - y^2)) \tag{4.43}$$

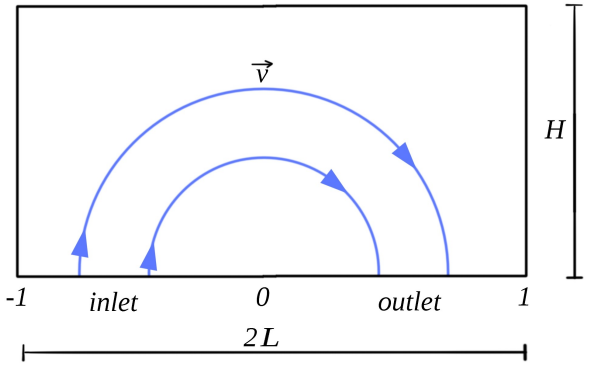


Figure 4.14: Smith-Hutton flow scheme. Geometry.

The following boundary conditions are applied:

1. **Left, right, and upper sides:** the Dirichlet boundary condition is applied, which establishes that all these nodes have a constant ϕ value. This value is defined by the following equation:

$$\phi = 1 - tanh(10) \tag{4.44}$$

2. **Lower side:** this side is divided into two parts. The inlet part, $x \in [-1, 0]$, and the outlet part, $x \in [0, 1]$. The inlet side has a Dirichlet boundary condition, while the outlet side has a Neumann boundary condition. Both equations are presented below, respectively:

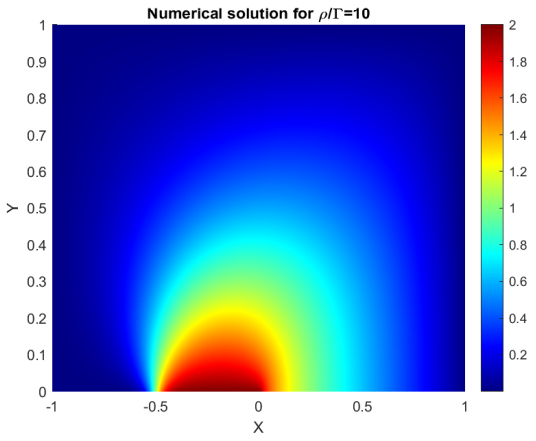
$$\phi = 1 + tanh(10(2x + 1)) \tag{4.45}$$

$$\frac{\partial \phi}{\partial y} = 0 \tag{4.46}$$

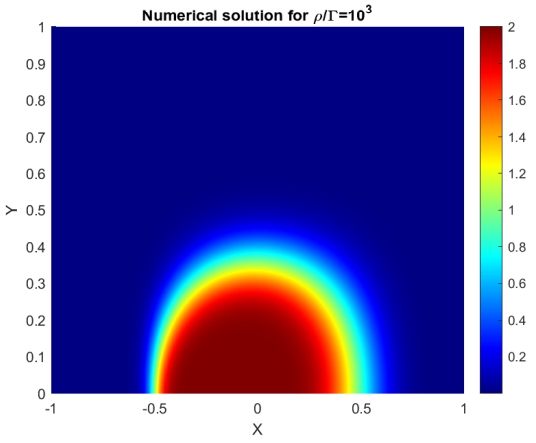
The results of this case will be compared with the ones provided by the CTTC, listed in the table below (*see Table (4.2)*). The figures represent the obtained results for the three different ρ/Γ relations with the UDS scheme.

Table 4.2: Expected results for the Smith-Hutton problem. Extracted from [8]

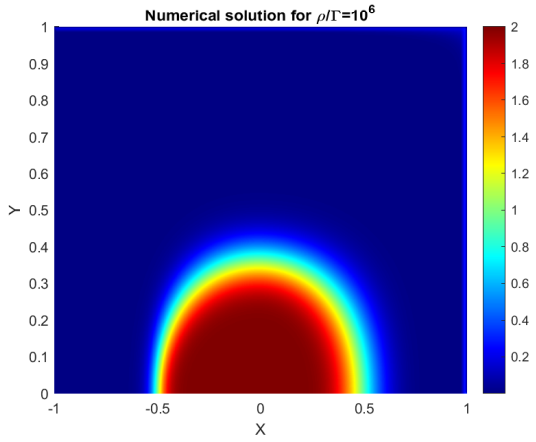
x-position [m]	$\rho/\Gamma = 10$	$\rho/\Gamma = 10^3$	$\rho/\Gamma = 10^6$
0.0	1.989	2.0000	2.000
0.1	1.402	1.9990	2.000
0.2	1.146	1.9997	2.000
0.3	0.946	1.9850	1.999
0.4	0.775	1.8410	1.964
0.5	0.621	0.9510	1.000
0.6	0.480	0.1540	0.036
0.7	0.349	0.0010	0.001
0.8	0.227	0.0000	0.000
0.9	0.111	0.0000	0.000
1.0	0.000	0.0000	0.000



(a) ρ/Γ =10



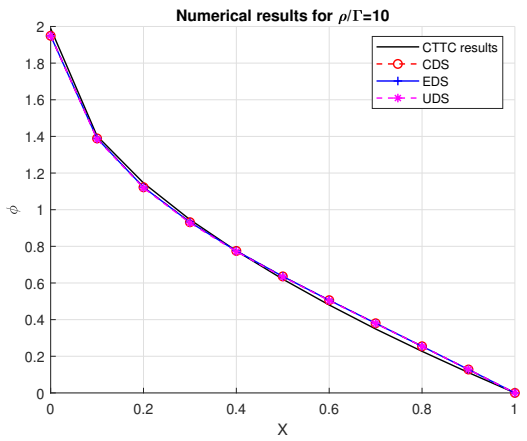
(b) $\rho/\Gamma = 10^3$



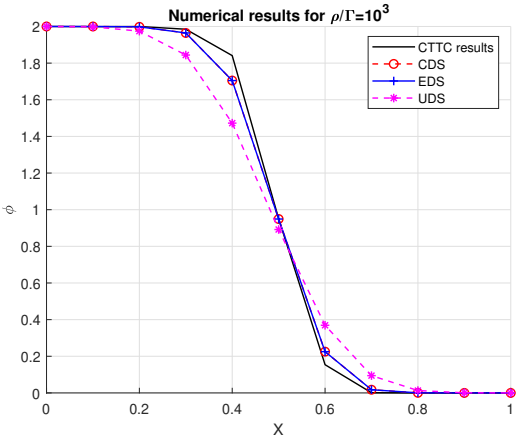
(c) $\rho/\Gamma = 10^6$

Figure 4.15: Numerical solution for different ρ/Γ values and 100×100 mesh with UDS

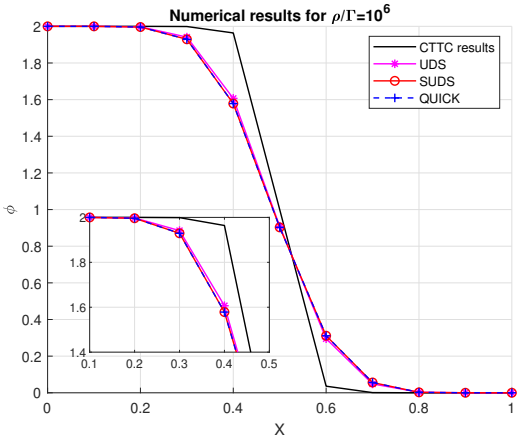
To compare the obtained numerical results with the reference values, the following plots have been done:



(a) ρ/Γ =10



(b) $\rho/\Gamma = 10^3$



(c) $\rho/\Gamma = 10^6$

Figure 4.16: Comparison between provided and obtained solutions for different ρ/Γ values and 100×100 mesh

4.4 Discussion

For this chapter, it can be concluded that the code has been successfully verified since the numerical results are according to the analytical solutions.

It is interesting to remark that all the tested convective schemes have successfully resulted in proper results. First-order schemes have induced more error than second-order schemes, and high-resolution schemes provided better results with coarser meshes than second-order schemes with finner meshes.

All the benchmark problems have been tested with the same algorithm, and the only difference between them is the boundary conditions of each particular case. For this reason, it can be concluded that the code is working perfectly and it is capable to deal with any convection-diffusion problem.

Chapter 5

The Fractional Step Method

The FSM is usually used to solve the incompressible Navier-Stokes equations due to its good performance, even with turbulence phenomena, and the simplicity of its code. It is commonly known as a projection method since its use of an intermediate velocity. This velocity is an approximate solution of the momentum equations but it cannot satisfy the incompressibility constraint. For this reason, the pressure Poisson equation is also used, which determines the minimum perturbation that will accomplish the incompressibility of the predictor velocity [10].

5.1 Mathematical formulation

Before diving into the subject, the previous step is to discuss the Helmholtz-Hodge theorem. It states as follows:

A given vector field $\vec{\omega}$, defined in a bounded domain Ω with smooth boundary $\partial\Omega$, is uniquely decomposed in a pure gradient field and divergence-free vector parallel to $\partial\Omega$

$$\vec{\omega} = \vec{a} + \nabla\psi \tag{5.1}$$

where

$$\nabla \cdot \vec{a} = 0 \qquad \vec{a} \in \Omega \tag{5.2}$$

The theorem also applies for periodic inflow/outflow conditions.

Taking this theorem into consideration, let's start with the mathematical formulation. The first step is to consider both mass and linear momentum Navier-Stokes equations for incompressible and constant viscosity flows:

$$\nabla \cdot \vec{v} = 0 \tag{5.3}$$

$$\rho\frac{\partial \vec{v}}{\partial t} = R(\vec{v}) - \nabla p \tag{5.4}$$

where

$$\vec{v} = u\vec{i} + v\vec{j} + w\vec{k} \tag{5.5}$$

and $R(\vec{v})$ is introduced to simplify equations, considering both convection and diffusion terms. It stands for:

$$R(\vec{v}) = -(\rho\vec{v}\cdot\nabla)\vec{v} + \nabla\cdot(\mu\nabla\vec{v}) \tag{5.6}$$

Equation (5.3) must be verified at any time between t^n and t^{n+1}. Time integration of equations (5.3) and (5.4) results in:

$$\nabla\cdot\vec{v}^{n+1} = 0 \tag{5.7}$$

$$\rho\frac{\vec{v}^{n+1}-\vec{v}^n}{\Delta t} = R(\vec{v}^{\frac{n+1}{2}}) - \nabla p^{n+1} \tag{5.8}$$

From equation (5.8), two considerations can be done:

1. Helmholtz-Hodge theorem is applied:

$$\vec{v}^{n+1} = \vec{v}^P - \left(\frac{\Delta t}{\rho}\nabla p^{n+1}\right) \tag{5.9}$$

2. $R(\vec{v}^{\frac{n+1}{2}})$ is interpolated as follows:

$$R(\vec{v}^{\frac{n+1}{2}}) = \frac{3}{2}R(\vec{v}^n) - \frac{1}{2}R(\vec{v}^{n-1}) \tag{5.10}$$

Equation (5.8) can be rewritten:

$$\rho\frac{\vec{v}^P-\vec{v}^n}{\Delta t} = \frac{3}{2}R(\vec{v}^n) - \frac{1}{2}R(\vec{v}^{n-1}) \tag{5.11}$$

From this equation (5.11), the predictor velocity can be computed, and introducing the results into equation (5.9), the real velocity for the next time step can be computed, both computations are done node-by-node. The pressure map is still unsolved. To do it, the same equation (5.9) is taken and its divergence is computed:

$$\nabla\cdot\vec{v}^{n+1} = \nabla\cdot\vec{v}^P - \nabla\cdot\left(\frac{\Delta t}{\rho}\nabla p^{n+1}\right) \tag{5.12}$$

Applying here equation (5.7), the Poison equation for pressure is found:

$$\Delta p^{n+1} = \frac{\rho}{\Delta t}\nabla\cdot\vec{v}^P \tag{5.13}$$

With this Poison equation, the incompressibility of the flow is always considered, and with a Gauss-Seidel solver, the pressure map can be computed.

The whole FSM code can be summarised in the following lines:

1. Evaluation of $R(\vec{v}^n)$ with equation (5.6)
2. Evaluation of predictor velocity $\vec{v}^P$ with equation (5.11) node-by-node

3. Gauss-Seidel solver for the pressure field p^{n+1} with Poison equation (5.13)

4. Evaluation of real velocity $\vec{v}^{n+1}$ with equation (5.9) node-by-node

The code's simplicity mentioned before is demonstrated with the previous scheme. Nonetheless, a more in-depth study of some relevant aspects is ruled hereafter, and an extended algorithm is presented below.

5.2 Staggered meshes

Until this chapter, numerical discretization was done using non-overlapping control volumes with cell-centered nodes, where scalar variables such as pressure and density were stored together with vector velocities at the cell-centered nodes. Staggered meshes store the scalar variables at the cell-centered nodes, but velocities, or momentum variables, are stored at the cell faces.

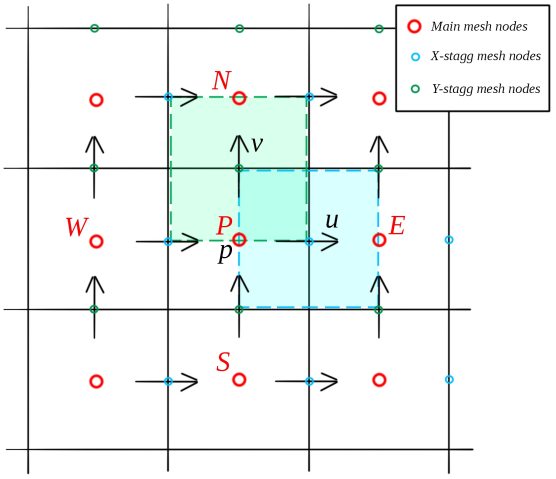


Figure 5.1: Staggered mesh

To better understand the need for a staggered mesh, it is mandatory to study the checkerboard problem and understand the possible decoupling between velocity and pressure.

5.2.1 The Checkerboard problem

Let's suppose an imaginary problem where only the next real velocity wants to be computed with the equation (5.9) node-by-node. Simplifying the problem to a 1D spatial discretization, the mentioned equation can be expressed in terms of current, west, and east nodes as follows:

$$u_P^{n+1} = u_P^p - \frac{\Delta t}{\rho}\left(\frac{p_E^{n+1} - p_W^{n+1}}{2\Delta x}\right) \tag{5.14}$$

As can easily be seen, the solution of the velocity at the current node is independent of the pressure of this node, which means that the convergence of the velocity can be obtained even with unphysical pressure distributions. For example, let's suppose the following pressure distribution:

$$p_{WW}^{n+1} = 10^3 \quad p_W^{n+1} = 0 \quad p_P^{n+1} = 10^3 \quad p_E^{n+1} = 0 \quad p_{EE}^{n+1} = 10^3$$

The velocity will converge since the condition $\nabla p^{n+1} = 0$ is accomplished, but the results are not coupled. This is why staggered meshes are needed; thus, the checkerboard problem is solved.

5.2.2 Staggered meshes in the FSM

The staggered mesh is a little bit trickier than the non-overlapping finite volume meshes seen before, but its implementation is quite easy. To study its implementation together with spatial discretization, the previous summarized code of four steps will be expanded.

1. The first step is to evaluate the $R(\vec{v}^n)$. For this purpose, the term is divided into $R(u^n)$ and $R(v^n)$. Each component of the velocity will be evaluated with the staggered mesh in the X and Y direction, respectively.

   (a) **x-staggered mesh:** equation (5.6) is particularized for the horizontal velocity and integrated over the x-staggered CV, and the Gauss theorem is applied. It results in the following equation:

   $$R(u^n)\Omega_{xP} = -conv + diff \tag{5.15}$$

   where Ω_{xP} stands for the x-staggered volume, and $conv$ and $diff$ stand for convective and diffusive terms, expressed as follows:

   $$conv = \dot{m}_e u_e - \dot{m}_w u_w + \dot{m}_n u_n - \dot{m}_s u_s \tag{5.16}$$

   $$diff = \mu_e \frac{u_E - u_P}{d_{PE}} A_e - \mu_w \frac{u_P - u_W}{d_{PW}} A_w + \mu_n \frac{u_N - u_P}{d_{PN}} A_n - \mu_s \frac{u_P - u_S}{d_{PS}} A_s \tag{5.17}$$

   Volumetric flow and velocities at the faces are the most tricky part of this method. The evaluation of the velocities at the CV faces is done by convective numerical schemes (CDS, UDS, SUDS, QUICK). For example, east face with CDS:

   $$u_e = u_P + \frac{d_{Pe}}{d_{PE}}(u_E - u_P) \tag{5.18}$$

   The evaluation of the volumetric flow at the east and west faces is made by the arithmetic mean of the horizontal velocities at the east and west nodes.

   $$\dot{m}_e = \frac{(\rho u)_E + (\rho u)_P}{2} A_e \tag{5.19}$$

   Finally, the evaluation of the volumetric flow at the north and south faces is made by the following equation:

   $$\dot{m}_n = (\rho v)_A A_{An} + (\rho v)_B A_{Bn} \tag{5.20}$$

   (b) **y-staggered mesh:** equation (5.6) is now particularized for the vertical velocity and integrated over the y-staggered CV, and the Gauss theorem is applied. It results in the same equation as

before but with the Y components:

$$R(v^n)\Omega_{yP} = -conv + diff \tag{5.21}$$

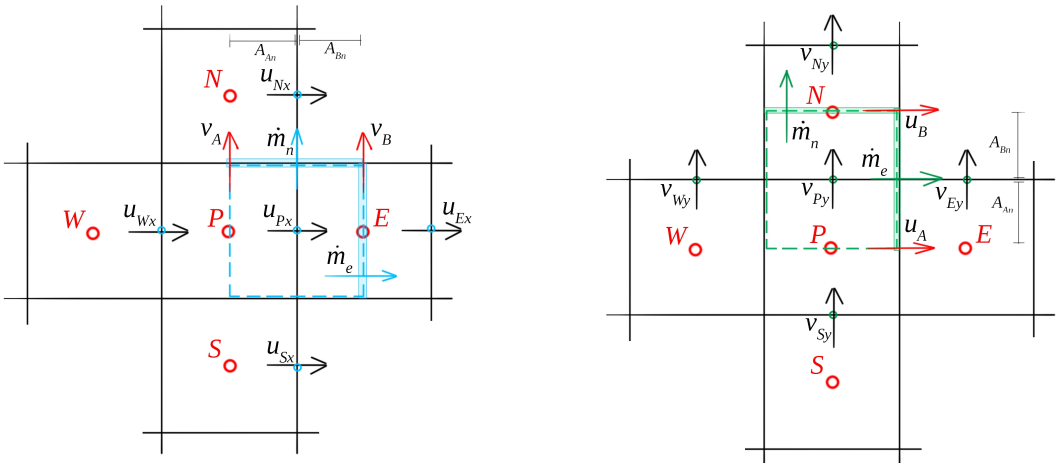
where Ω_{yP} stands for the y-staggered volume. Convective and diffusive terms are the same as before but with vertical velocities. Again, the evaluation of the velocities at the control volume faces is done by convective numerical schemes. Then, the evaluation of the volumetric flow at the north and south faces is made by the arithmetic mean of the vertical velocities at the north and south nodes:

$$\dot{m}_n = \frac{(\rho u)_N + (\rho u)_P}{2} A_n \tag{5.22}$$

And the evaluation of the volumetric flow at the east and west faces is made by the following equation:

$$\dot{m}_e = (\rho u)_A A_{Ae} + (\rho u)_B A_{Be} \tag{5.23}$$

The following scheme illustrates the previous distances, such as A_{An}, A_{Bn}, A_{Ae} or A_{Be} and where each value is evaluated, for x-staggered and y-staggered meshes:



(a) x-staggered mesh

(b) y-staggered mesh

Figure 5.2: Distances and values evaluation positions

2. Once $R(u^n)$ and $R(v^n)$ are computed with the x-staggered and y-staggered meshes, the predictory velocity can be computed node-by-node. As before, it is split into the horizontal and vertical velocities:

$$u^P = u^n + \frac{\Delta t}{\rho}\left(\frac{3}{2}R(u^n) - \frac{1}{2}R(u^{n-1})\right) \tag{5.24}$$

$$v^P = v^n + \frac{\Delta t}{\rho}\left(\frac{3}{2}R(v^n) - \frac{1}{2}R(v^{n-1})\right) \tag{5.25}$$

3. The pressure is computed with the Poison equation (5.13). After numerical integration and grouping of the different terms, the obtained equation is the following one. It must be noted that the pressure is

solved for the main mesh:

$$a_P p_P^{n+1} = a_E p_E^{n+1} + a_W p_W^{n+1} + a_N p_N^{n+1} + a_S p_S^{n+1} + b_P \tag{5.26}$$

where

$$a_E = \frac{A_e}{d_{PE}} \qquad a_W = \frac{A_w}{d_{PW}} \qquad a_N = \frac{A_n}{d_{PN}} \qquad a_S = \frac{A_s}{d_{PS}} \tag{5.27}$$

$$a_P = a_E + a_W + a_N + a_S \tag{5.28}$$

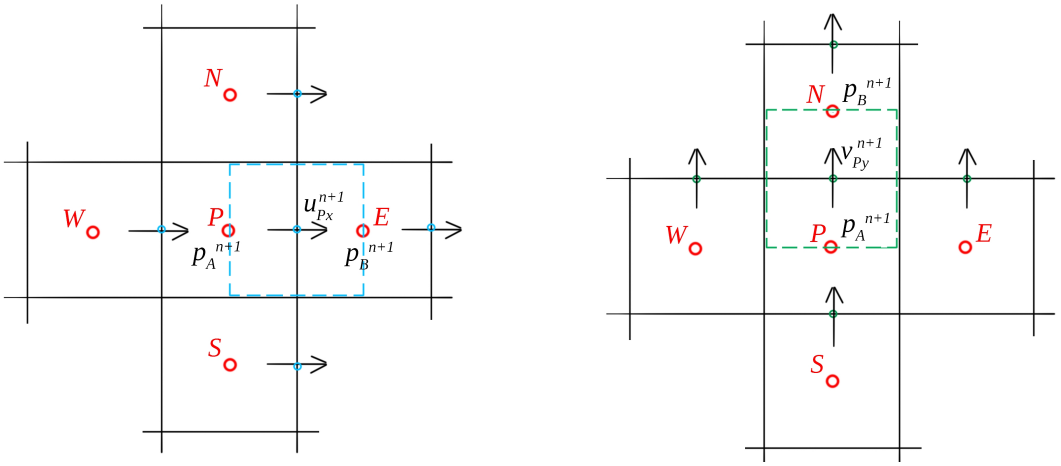
$$b_P = \frac{-1}{\Delta t}\left((\rho u^P)_e A_e - (\rho u^P)_w A_w + (\rho v^P)_n A_n - (\rho v^P)_s A_s\right) \tag{5.29}$$

4. Finally, the next velocity $\vec{v}^{n+1}$ can be computed node-by-node. Once again, the velocity is split into horizontal and vertical components, and each component is solved with the x-staggered and y-staggered meshes, respectively. The equations are the following:

$$u_P^{n+1} = u_P^P - \frac{\Delta t}{\rho}\left(\frac{p_B^{n+1} - p_A^{n+1}}{d_{BA}}\right) \tag{5.30}$$

$$v_P^{n+1} = v_P^P - \frac{\Delta t}{\rho}\left(\frac{p_B^{n+1} - p_A^{n+1}}{d_{BA}}\right) \tag{5.31}$$

The following scheme illustrates where the previous pressures are located, and the distances for x-staggered and y-staggered meshes:



(a) x-staggered mesh (b) y-staggered mesh

Figure 5.3: Pressure evaluation positions

5.2.3 Time step choice

Since the time integration is an explicit one, the time step must be low enough to ensure the stability of the simulation. To do so, the time step choice is subject to the Courant-Friedrich-Levy condition (CFL condition), which stands for the minimum time step to maintain the simulation stability. To do so, the

minimum convective and diffusive time steps are computed:

$$\Delta t_c = min\left(0.35\frac{\Delta x}{|\vec{v}|}\right) \tag{5.32}$$

$$\Delta t_d = min\left(0.2\frac{\rho\Delta x^2}{\mu}\right) \tag{5.33}$$

The minimum of these is used for the simulation. It must be noted that the time step is recomputed in each time step:

$$\Delta t = min(\Delta t_c, \Delta t_d) \tag{5.34}$$

5.3 Code structure

The general FSM algorithm is the following one. The same code can be used to solve the different cases by changing boundary conditions and adding additional equations to solve energy, or transport species, if necessary.

1. Input data:
   (a) Physical data, such as geometry, initial values, and physical properties.
   (b) Numerical data, such as convergence parameter, mesh size, etc.
2. Previous calculations: vector definitions and mesh generation. Distances and areas for the x-staggered and y-staggered meshes are computed, and the *Block-off Method* is applied, if necessary.
3. Initial maps: $\vec{v}$, $R(\vec{v}^{n-1})$, and pressure map.
4. Discretization coefficients evaluation: a_E, a_W, a_N, a_S, a_P
5. Time-convergence starting point: $t = t + \Delta t$
   (a) CFL condition: minimum time step evaluation.
   (b) $R(\vec{v}^n)$ evaluation
   (c) Intermediate velocity evaluation $\vec{v}^P$ node-by-node.
   (d) b_P coefficient evaluation with the new time step and intermediate velocities.
   (e) Pressure resolution with Gauss-Seidel solver:

$$a_P p = a_E p_E^{est} + a_W p_W^{est} + a_N p_N^{est} + a_S p_S^{est} + b_P \tag{5.35}$$

   (f) Next velocity evaluation $\vec{v}^{n+1}$ node-by-node.
   (g) Convergence analysis: if the result converges, $|max(p^{n+1} - p^n)| < \delta$ the code goes to step 6. If not, the code updates data, $\vec{v}^{n-1} = \vec{v}^n$, $\vec{v}^n = \vec{v}^{n+1}$, $R(\vec{v}^n) = R(\vec{v}^{n+1})$, $p^n = p^{n+1}$, and goes back to step $5a$.

6. Time convergence analysis: $t \geqslant t_{final}$
7. Final computations and results printing

5.4 Code verification

The code verification is done by four different benchmark problems. Even though each problem has different characteristics, the main code is almost the same since all problems implement the FSM.

Particular treatment of the boundary conditions is done for each case. The following sections present these boundary conditions and the obtained results for each case, along with the reference results and their comparison.

5.4.1 Lid-Driven Cavity

The first benchmark problem is the Lid-Driven Cavity case. Let's suppose that an incompressible and laminar flow is enclosed in a square cylinder [11]. The following boundary conditions are applied:

1. **Left, right, and lower sides:** these are considered static walls, and the Neumann boundary condition for pressure is applied to the vertical component at the lower side, and to the horizontal component at the other sides. Regarding velocities, both horizontal and vertical components take null values due to the no-slip condition:

$$\frac{\partial p}{\partial y} = 0 \quad \text{and} \quad \frac{\partial p}{\partial x} = 0 \tag{5.36}$$

$$\vec{v} = 0 \tag{5.37}$$

2. **Upper side:** this side is also a wall, so the same Neumann boundary condition is applied for the pressure field as it was for the lower side. Regarding velocities, this is a moving wall, so horizontal velocity takes a constant known value:

$$u = u_{ref} \tag{5.38}$$

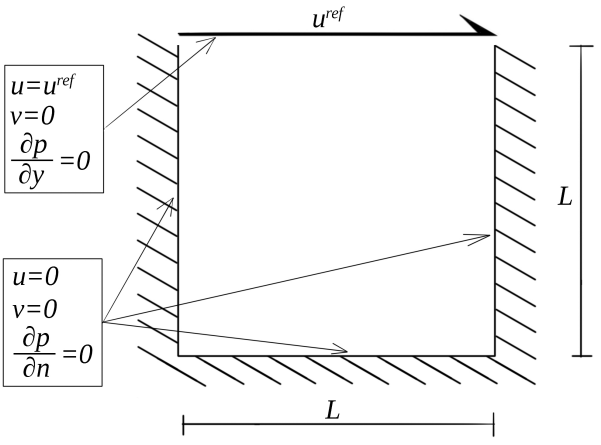


Figure 5.4: Lid-Driven Cavity scheme. Geometry and boundary conditions

The problem will be solved with a uniform mesh and constant known thermophysical properties for different meshes, convective numerical schemes, and Reynolds values. The following list sums up the simulation parameters for the obtained results:

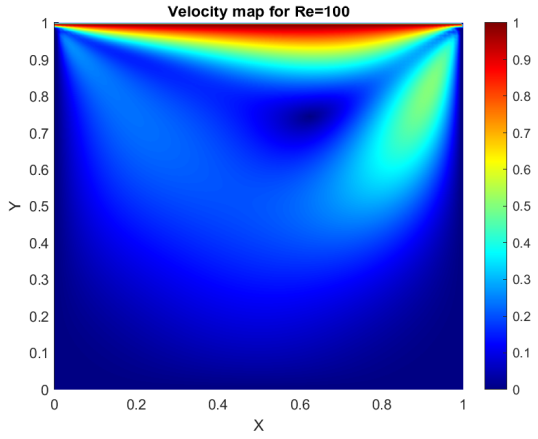
1. **50×50 mesh:** CDS, UDS and SUDS *(SUDS only for Reynolds 100, 1000, and 5000)*
2. **100×100 mesh:** CDS
3. **Reynolds numbers:** 100, 400, 1000, 3200, 5000, 7500 and 10000
4. **Steady-state convergence parameter:** 10^{-4}
5. **Convergence parameter:** 10^{-6}

The reasons why CDS has been chosen to be the main scheme are the following:

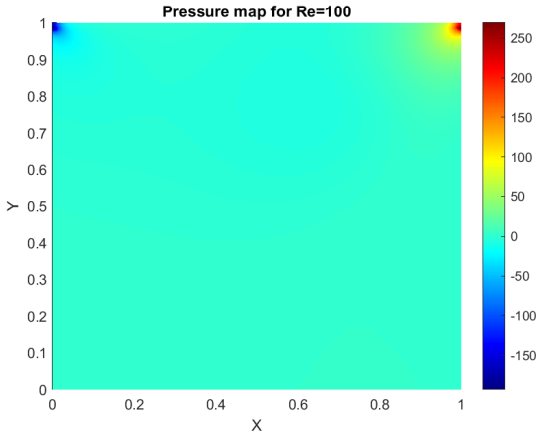
1. CDS, as said in its respective chapter, is a 2nd-order accurate scheme. It presents stability problems but these problems are solved thanks to the CFL condition. For this reason, CDS becomes a 2nd-order accurate and stable scheme.
2. UDS is a much more stable scheme but it is only 1st-order accurate. For this reason, with the same mesh, CDS results are expected to be much more accurate than the ones obtained via UDS.
3. SUDS takes both good parts of the previous schemes, being a stable and 2nd-order accurate scheme. In return, its computational cost is significantly increased. For the coarse mesh, it returns results with an accuracy close to the CDS results with the finner mesh, but the computational time is much greater. For this reason, SUDS simulations with a fine mesh could take days.

For all the reasons mentioned above, CDS simulations will be ruled with 50×50 and 100×100 meshes. UDS simulations will be done for 50×50, as well as SUDS simulations, but this last scheme will only be applied to low, medium, and high Reynolds, to illustrate the evolution of the accuracy with the Reynolds number.

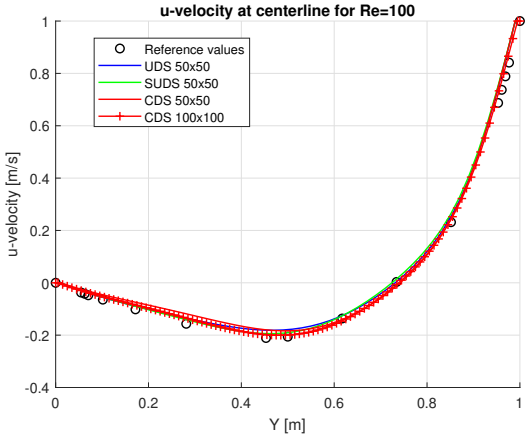
The obtained results for different Reynolds are illustrated in the following pages, together with the reference results [11]. At the end of the simulation results, a table with the computational times and time convergences of every simulation can be found (*see Table (5.1)*):



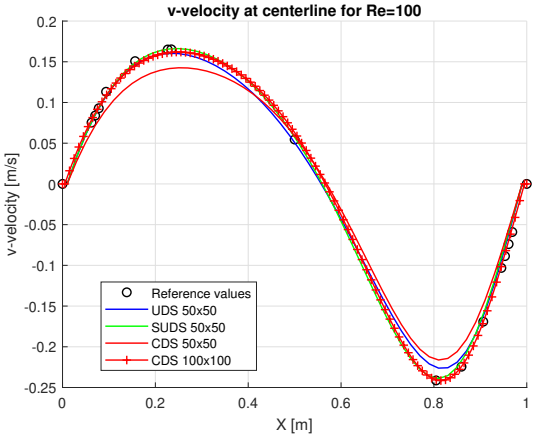
(a) Velocity map. CDS and 100×100 mesh



(b) Pressure map. CDS and 100×100 mesh



(c) u-velocity comparison



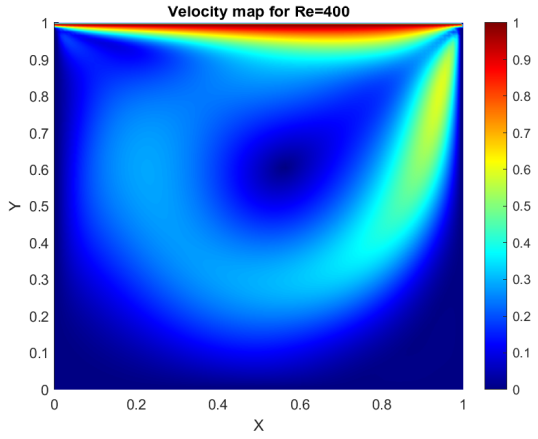
(d) v-velocity comparison

Figure 5.5: Numerical solutions for Re=100

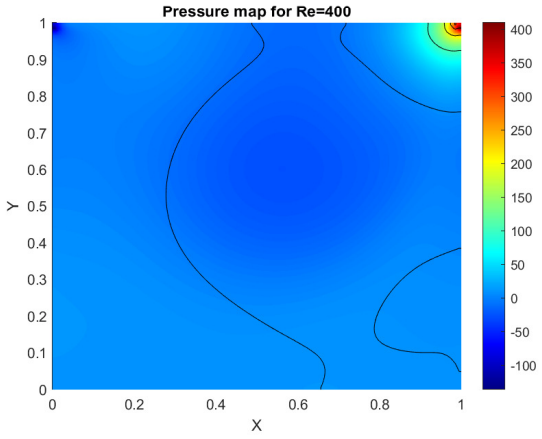
From these results, some interesting conclusions can be extracted.

The first one is the singularity points in the pressure map. The whole cavity is considered to be at a null pressure value, but the upper left and right corners are taking high positive and negative pressure values. These singularity points occur due to the combination of two factors, which are the flow separation and the boundary conditions. When the fluid near the corners of the cavity encounters an abrupt change in direction, it tends to separate from the surface, creating a region of recirculation or stagnation. This flow separation leads to the formation of vortices or eddies, which are commonly observed in the corners of the cavity. The presence of these vortices near the corners results in localized pressure variations. In the upper left corner, for example, the flow separation creates a region of low-pressure or suction, causing the pressure to drop to negative values. Conversely, in the upper right corner, the flow separation results in a region of high pressure, leading to positive pressure values.

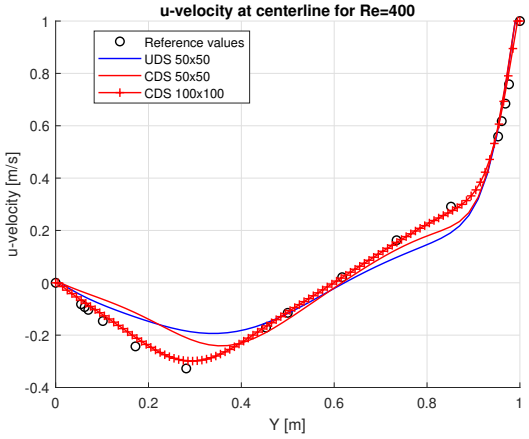
The second one is the mesh study. It is clearly visible how almost the same results are obtained for the different schemes and meshes. CDS with the coarse mesh, and UDS provide some error in the vertical velocity, but the SUDS results with the coarse mesh, and CDS results with the fine mesh appear to be the same.



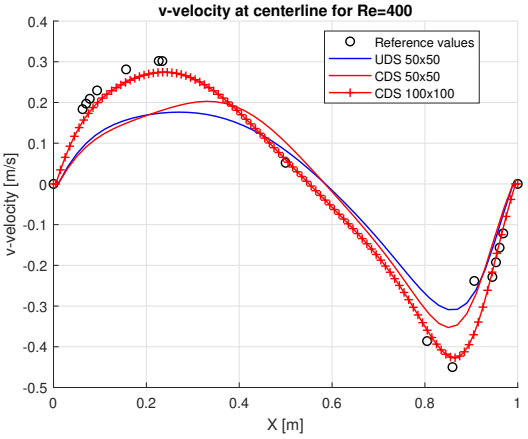
(a) Velocity map. CDS and 100×100 mesh



(b) Pressure map. CDS and 100×100 mesh



(c) u-velocity comparison

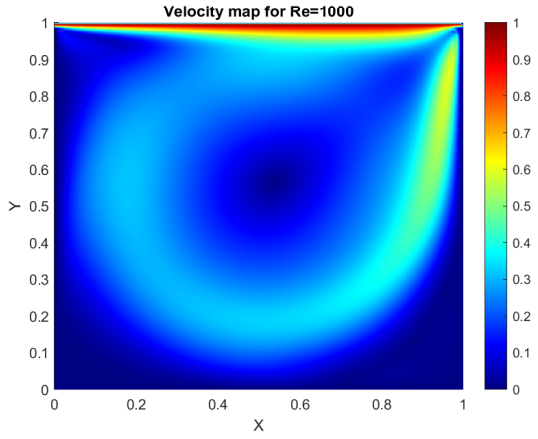


(d) v-velocity comparison

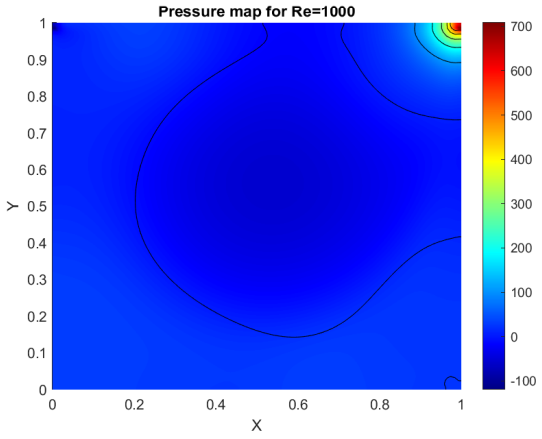
Figure 5.6: Numerical solutions for Re=400

The Reynolds has slightly increased. The flow has developed a greater recirculation behavior. The pressure map seems to be taking null pressure values for all the fluid domain. About the singularity points, the suction corner has moderated its behavior, equalizing its pressure value with the rest of the cavity, but the high-pressure corner has increased its value, becoming more aggressive.

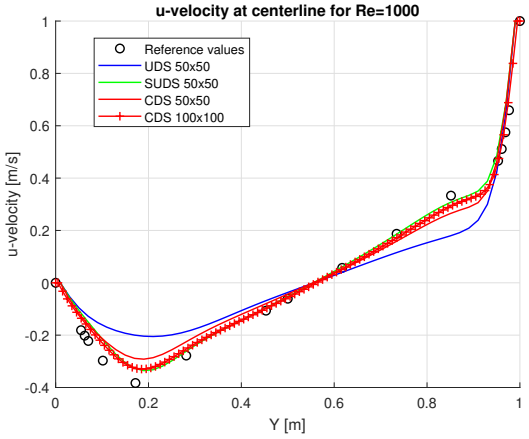
About the meshes, in this case, a differentiation between results starts to be visible. UDS and CDS results for the coarse mesh are differing more with respect to the reference ones. CDS results with the fine mesh are still pretty accurate. It must be kept in mind that the reference results are extracted from a 129×129 mesh, so there will always be a certain error.



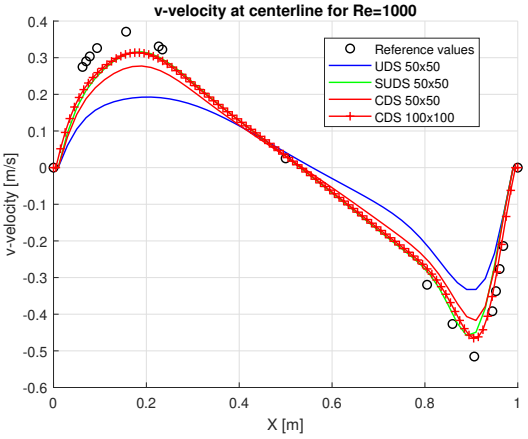
(a) Velocity map. CDS and 100×100 mesh



(b) Pressure map. CDS and 100×100 mesh



(c) u-velocity comparison

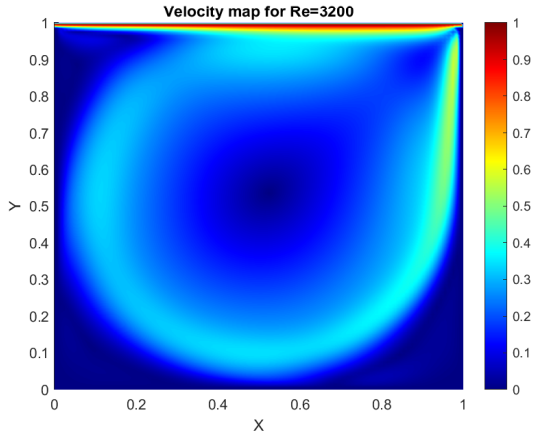


(d) v-velocity comparison

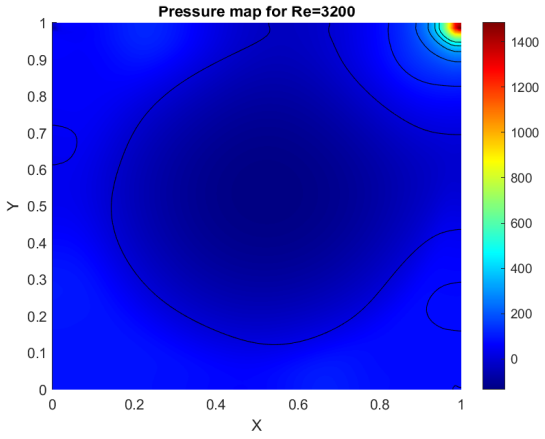
Figure 5.7: Numerical solutions for Re=1000

The recirculation behavior is still increasing, thus the recirculation zones at the corners are getting smaller. The low-pressure singularity point has almost disappeared, and the high-pressure one is getting higher, as long as the Reynolds increases.

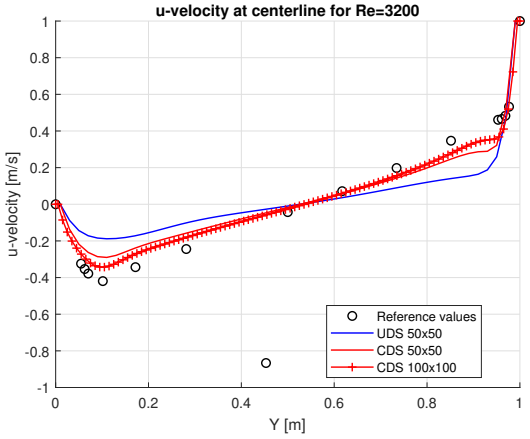
About the mesh and schemes study, UDS has no longer a relevant accuracy compared to the CDS and SUDS schemes. There are peaks of error up to 50% in the UDS. CDS results with a coarse mesh are starting to be inaccurate, and SUDS results with coarse mesh together with CDS results with fine mesh are still almost the same, and very similar to the reference ones, taking into consideration the base error due to the mesh difference between reference and numerical results.



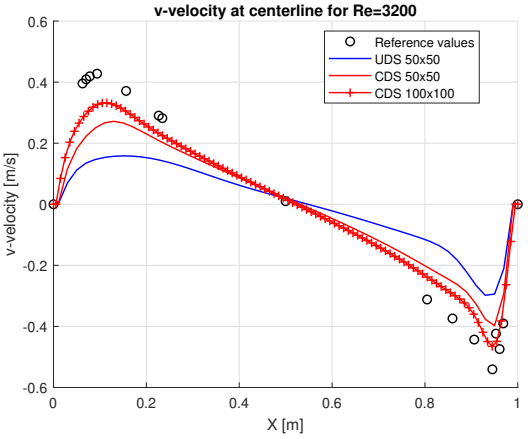
(a) Velocity map. CDS and 100×100 mesh



(b) Pressure map. CDS and 100×100 mesh



(c) u-velocity comparison

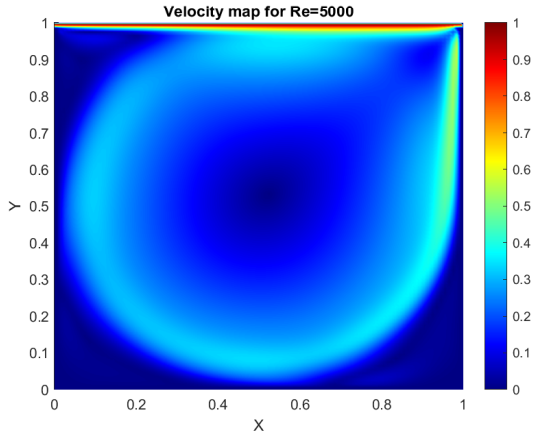


(d) v-velocity comparison

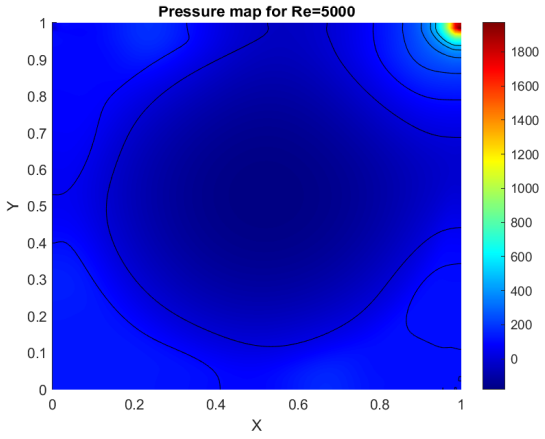
Figure 5.8: Numerical solutions for Re=3200

At this point, a relation between the Reynolds number and the obtained results can be established. For the velocity map, the recirculation is getting closer to the walls and the null-velocity zone at the center of the cavity is getting wider. The high-pressure zone is also getting higher with the Reynolds, and the suction point has already disappeared.

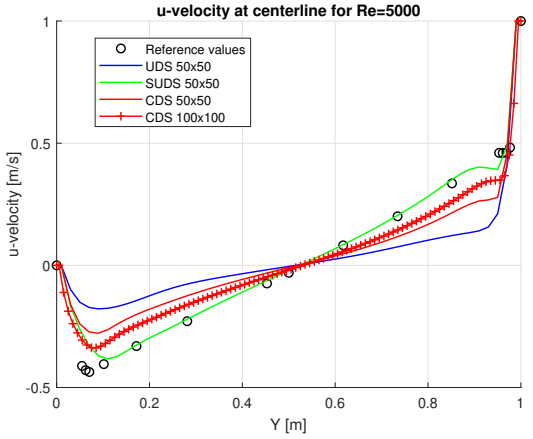
The error between reference results and UDS ones has increased too. This is no longer an accurate scheme, and it is only valid for the lowest Reynolds. CDS results in a coarse mesh are also differing from the results, and CDS results obtained with the fine mesh are also starting to differ from the reference ones. The nominal error due to the numerical simulations together with the error due to the mesh size is starting to move away the obtained results from the desired ones.



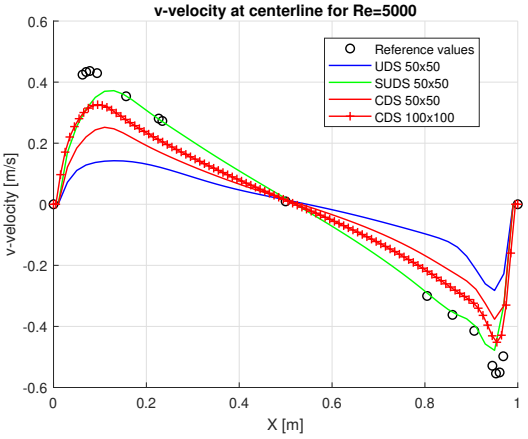
(a) Velocity map. CDS and 100×100 mesh



(b) Pressure map. CDS and 100×100 mesh



(c) u-velocity comparison

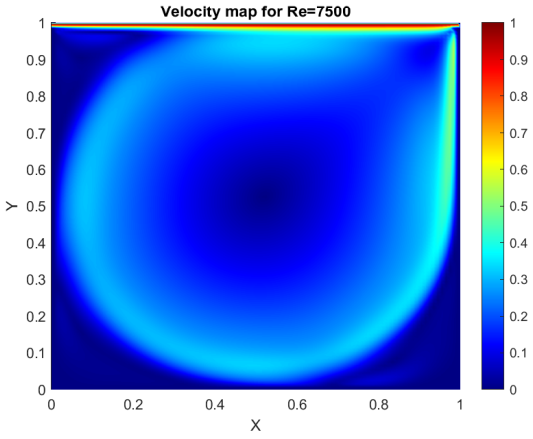


(d) v-velocity comparison

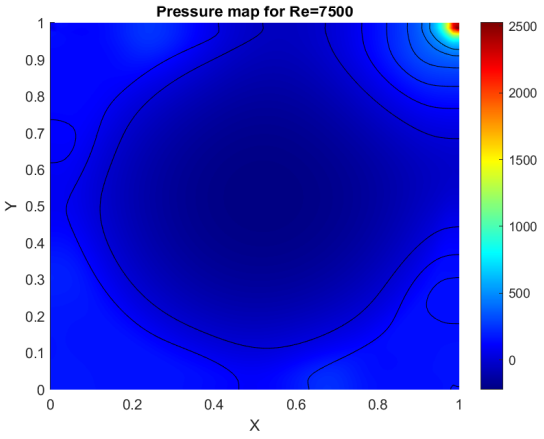
Figure 5.9: Numerical solutions for Re=5000

The vertical recirculation stream of the right wall is getting narrower with the Reynolds increase. The same stagnation point in the upper right corner is found, with a greater pressure value, and the center of the cavity is beginning to be a low-pressure zone.

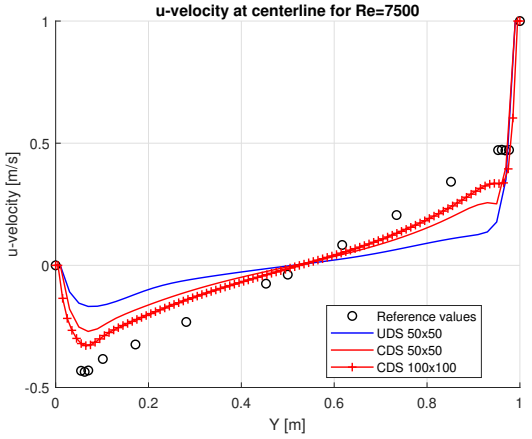
Following the tendency of the past simulations, UDS is inducing very big errors with respect to the reference results. CDS for a coarse mesh is also no longer a credible result and the same scheme for fine mesh has also decreased its accuracy. Only SUDS with coarse mesh is attached to the reference results. Despite that, the computational cost makes it a nonsuitable option to simulate with even higher Reynolds.



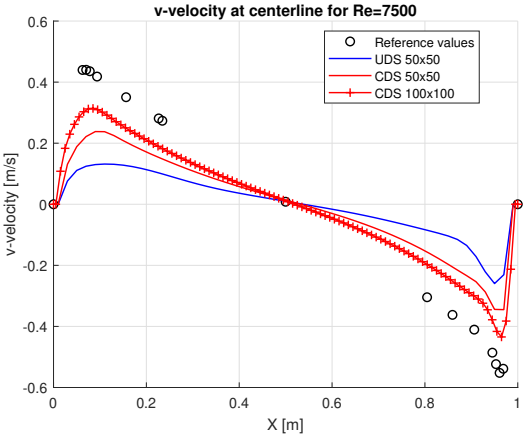
(a) Velocity map. CDS and 100×100 mesh



(b) Pressure map. CDS and 100×100 mesh



(c) u-velocity comparison

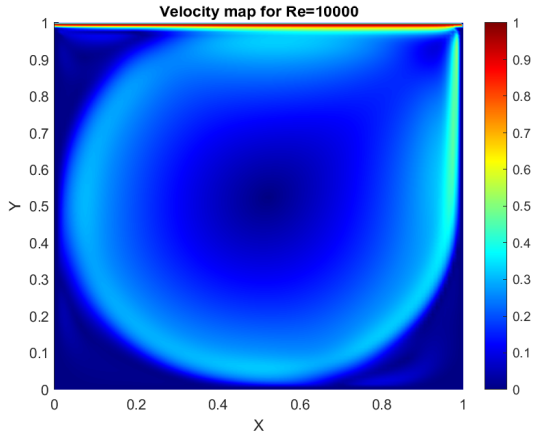


(d) v-velocity comparison

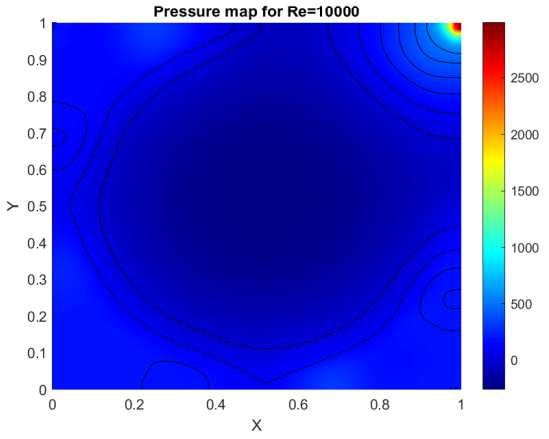
Figure 5.10: Numerical solutions for Re=7500

No differences between the previous simulation can be appreciated for this new Reynolds value. The narrowing of the vertical recirculation in the right wall is continuing, and the singularity point of the pressure map is taking even higher values.

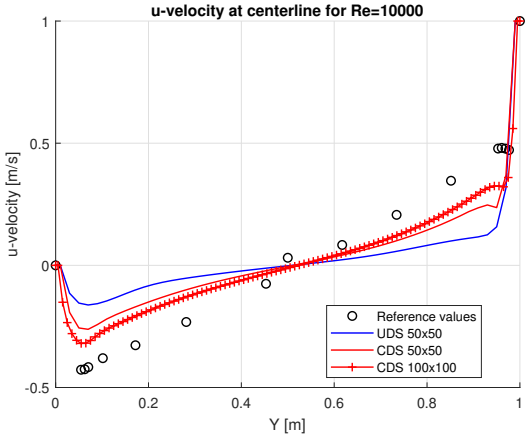
About the schemes and meshes, as before, UDS and CDS with coarse mesh are no longer a reliable results, and the error between reference results and CDS ones with fine mesh is getting higher as the Reynolds increase.



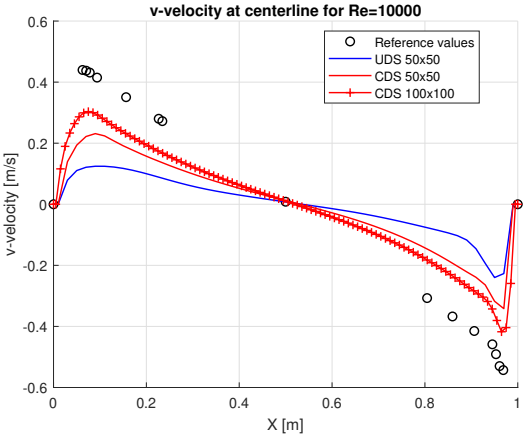
(a) Velocity map. CDS and 100×100 mesh



(b) Pressure map. CDS and 100×100 mesh



(c) u-velocity comparison



(d) v-velocity comparison

Figure 5.11: Numerical solutions for Re=10000

Finally, the last Reynolds number illustrates the narrowest vertical recirculation and the highest value of the pressure singularity point.

The results obtained for the horizontal and vertical velocities are very similar to the ones obtained in the previous simulation. The highest error between reference results and CDS ones with fine mesh is the highest of all the previous ones.

Table 5.1: Computational and time convergence for the different simulations

Reynolds	**Computational time [s]**				**Convergence time [s]**
	UDS 50X50	**SUDS 50X50**	**CDS 50X50**	**CDS 100X100**	**(aproximation)**
100	356	19500	4326	55860	15
400	241	-	7394	11640	30
1000	285	13560	3182	16080	45
3200	323	-	5996	31380	160
5000	364	24210	1817	38100	180
7500	424	-	1094	39060	200
10000	467	-	1273	50040	230

5.4.2 Differential Heated Cavity

The second benchmark problem is the Differential Heated Cavity case. Let's suppose the same incompressible and laminar flow which is enclosed in a square cavity. This time, the goal of the case is to obtain the velocity, pressure, and temperature maps generated by natural convection [12].

The following boundary conditions are applied:

1. **Left side:** this side is considered to be a constant temperature static wall. For this reason, a constant T_0 is applied to all the nodes of the wall, and null velocities due to the no-slip condition:

$$\vec{v} = 0 \tag{5.39}$$

2. **Right side:** this side is also considered to be a constant temperature static wall. For this reason, a constant T_1 is applied to all the nodes of the wall, and also null velocities.

3. **Upper and lower sides:** these are considered as adiabatic walls, and the Neumann boundary condition is applied for the temperature. Regarding velocities, both horizontal and vertical components take null values due to the no-slip condition:

$$\frac{\partial T}{\partial y} = 0 \tag{5.40}$$

$$\vec{v} = 0 \tag{5.41}$$

The following figure illustrates the geometry and boundary conditions of the problem:

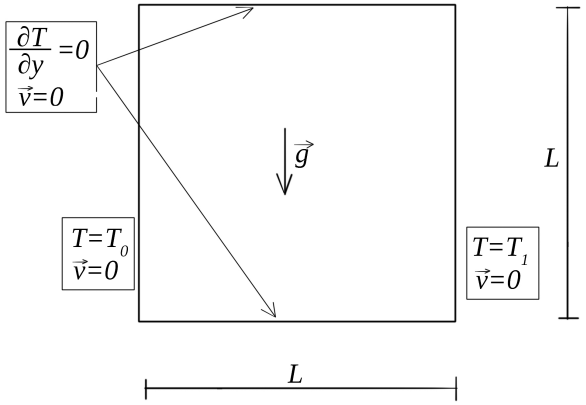


Figure 5.12: Differential Heated Cavity scheme. Geometry and boundary conditions

As this case studies natural convection, a gravity term is added. In addition, an extra Navier-Stokes equation must be added and, consequently, solved. This equation at hand is the energy equation. To better understand the influence of these new parameters, a mathematical formulation section is introduced below before the results are extracted.

Mathematical formulation

As said before, gravitational forces are now considered. For this reason, the Boussinesq approximation is introduced in the vertical component of the linear momentum equation. This approximation includes a density variation as a function of the temperature flow and the expansion coefficient κ:

$$\rho = \rho_0(1 - \kappa(T - T_0)) \tag{5.42}$$

The new mass, linear momentum, and energy Navier-Stokes equations for incompressible and constant viscosity flows are the following:

$$\nabla \cdot \vec{v} = 0 \tag{5.43}$$

$$\rho_0 \frac{\partial \vec{v}}{\partial t} = R(\vec{v}) - \nabla p \tag{5.44}$$

$$\frac{\partial T}{\partial t} = R(T) \tag{5.45}$$

where $R(u)$ does not change from the general mathematical formulation, but $R(v)$ and $R(T)$ are modified. They stand for:

$$R(v) = -(\rho_0 \vec{v} \cdot \nabla)\vec{v} + \nabla \cdot (\mu \nabla \vec{v}) - \rho_0(1 - \kappa(T - T_0))g \tag{5.46}$$

$$R(T) = \Delta T - (\vec{v} \cdot \nabla)T \tag{5.47}$$

On one hand, equation (5.46) is integrated as it was done in the general method. The resultant equation is the same and only adds the Boussinesq equation at the end. On the other hand, equation (5.47) is integrated in the same way. The result is the following:

$$R(T^n)\Omega_P = -conv + diff \tag{5.48}$$

where Ω_P stands for the volume, and $conv$ and $diff$ stand for convective and diffusive terms, expressed as follows:

$$conv = \dot{m}_e T_e - \dot{m}_w T_w + \dot{m}_n T_n - \dot{m}_s T_s \tag{5.49}$$

$$diff = \frac{\lambda}{\rho_0 c_p}\left(\frac{T_E - T_P}{d_{PE}}A_e - \frac{T_P - T_W}{d_{PW}}A_w + \frac{T_N - T_P}{d_{PN}}A_n - \frac{T_P - T_S}{d_{PS}}A_s\right) \tag{5.50}$$

Once $R(T^n)$ is computed, the new temperature can be computed node-by-node by time integration of equation (5.45):

$$T^{n+1} = T^n + \frac{\Delta t}{\rho_0 c_p}\left(\frac{3}{2}R(T^n) - \frac{1}{2}R(T^{n-1})\right) \tag{5.51}$$

Finally, another time convergence criterion is added to the CFL condition:

$$\Delta t_T = min\left(0.2\frac{\Delta x^2}{\frac{\lambda}{\rho_0 c_p}}\right) \tag{5.52}$$

The results are presented on the following pages. To compare them with the reference results, some additional

code lines must be added to compute the Nusselt matrix. The Nusselt number is a dimensionless number, as well as Reynolds or Péclet. It establishes the ratio between convective and conductive heat transfer, and it is computed by means of dimensionless parameters, denoted by * superscript:

$$\overline{Nu} = \frac{1}{x_L^* - x_0^*} \sum_i Nu_x \Delta x^* \tag{5.53}$$

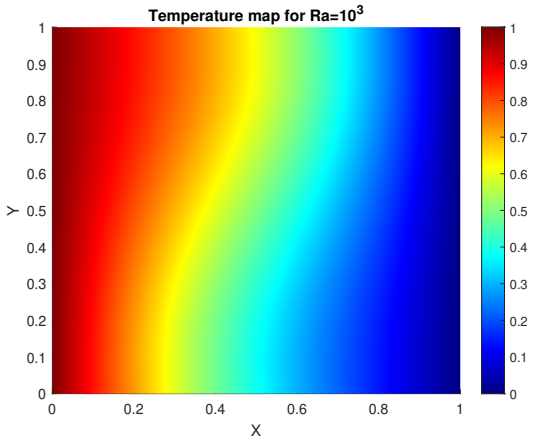
The previous equation computes the average Nusselt number of the cavity. x_L^* and x_0^* stand for the final and initial horizontal dimensionless positions, which are L and 0. Δx^* is the dimensionless distance between contiguous nodes, and Nu_x is the Nusselt number at a certain vertical line, and it is computed as follows:

$$Nu_x = \frac{1}{y_L^* - y_0^*} \sum_j \left(u_P^* T_P^* - \frac{T_E^* - T_P^*}{2d_{EP}} + \frac{T_P^* - T_W^*}{2d_{PW}} \right) \Delta y^* \tag{5.54}$$

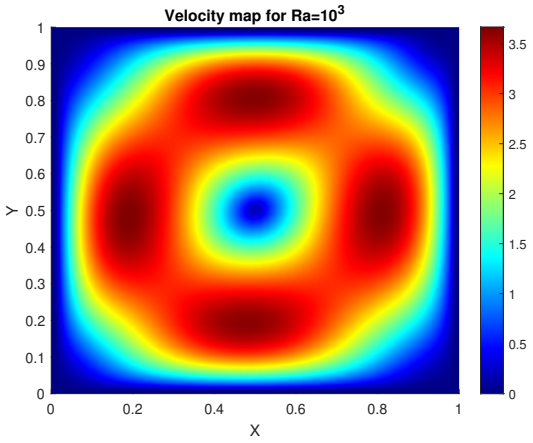
The dimensionless parameters are listed below:

$$x^* = \frac{x}{L} \quad y^* = \frac{y}{L} \quad T^* = \frac{T - T_0}{T_1 - T_0} \quad u^* = \frac{u\lambda}{\rho_0 c_p L} \tag{5.55}$$

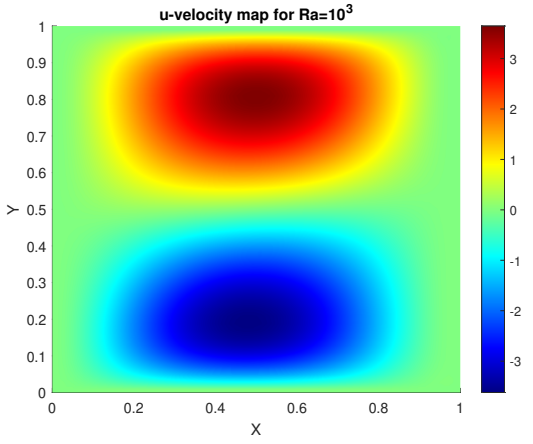
The problem will be solved with a uniform mesh and constant known thermophysical properties for different Rayleigh numbers, except for the density, which is considered to be variable for the gravitational terms. The chosen convective numerical scheme is CDS for both velocities and temperatures. In the end, two tables comparing reference and numerical results can be found.



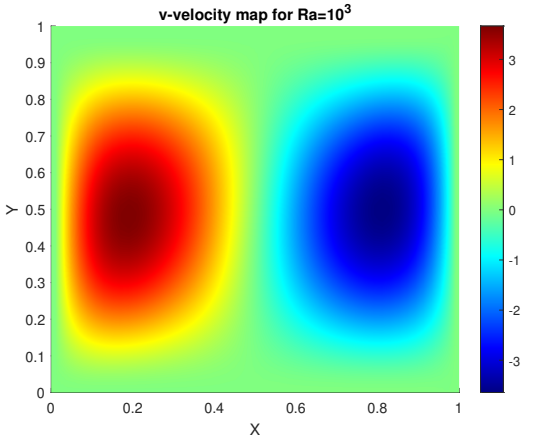
(a) Temperature map. CDS and 50×50 mesh



(b) Velocity map. CDS and 50×50 mesh



(c) Horizontal velocity map.

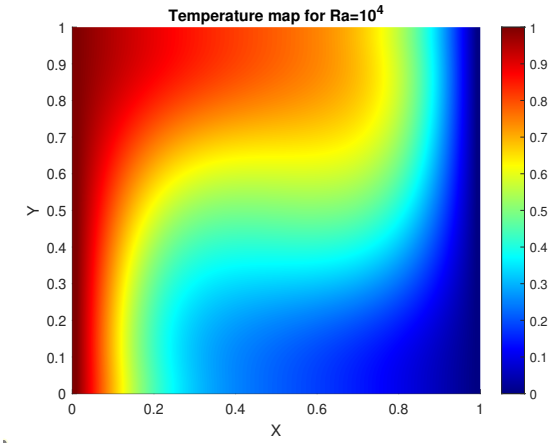


(d) Vertical velocity map.

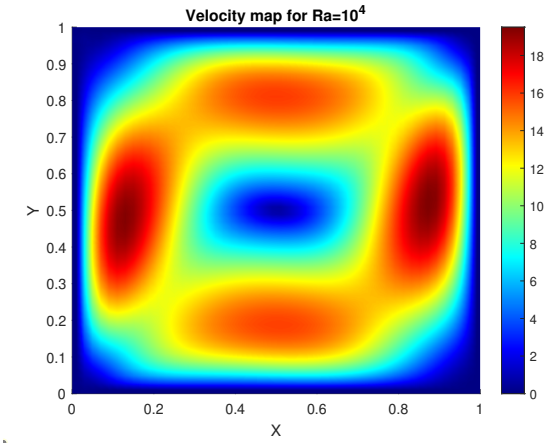
Figure 5.13: Numerical solutions for Ra=10^3

The first group of figures illustrates the results obtained for the first Rayleigh number. The temperature map (*see Figure (5.13a)*) shows how the hot fluid tends to ascend and the cold one tends to descend, due to gravitational forces and density variations.

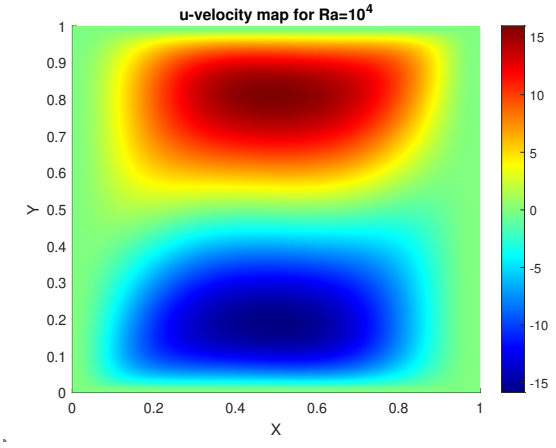
This behavior translated into velocities results in a constant circulation of the fluid, as can be seen in the velocity map (*see Figure (5.13b)*). This absolute velocity map is obtained due to the horizontal and vertical components, illustrated in the subfigures below (*see Figures (5.13c) and (5.13d)*). From the velocity map, it can be seen how the four corners are not affected by velocity and, thus stagnation points are found, likewise the center of the cavity. In addition, is clearly visible how, due to the no-slip condition, the nodes at the walls are taking null velocity values.



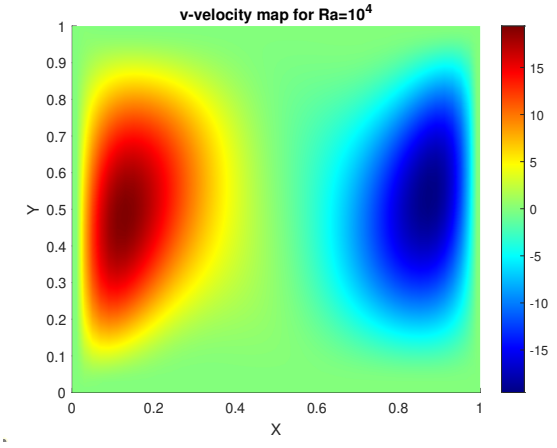
(a) Temperature map. CDS and 50×50 mesh



(b) Velocity map. CDS and 50×50 mesh



(c) Horizontal velocity map.



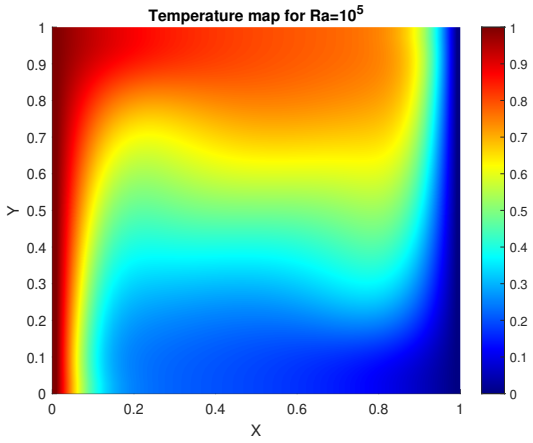
(d) Vertical velocity map.

Figure 5.14: Numerical solutions for Ra=10^4

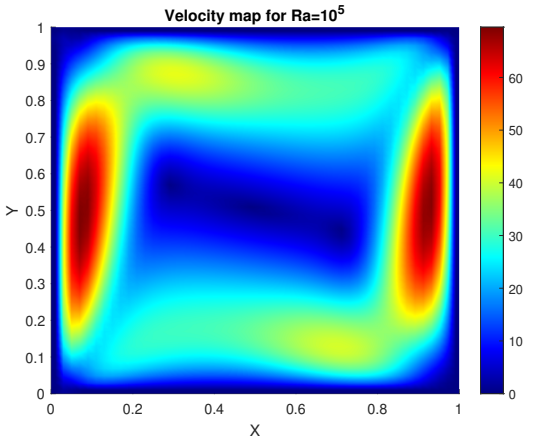
In this second simulation, the Rayleigh number has increased. This increase is translated into a more accentuated behavior of the temperatures and velocities.

The temperature map shows how the hot fluid has started to occupy a greater upper area, as well as the cold fluid for the lower area. The center of the cavity seems to be at a constant temperature.

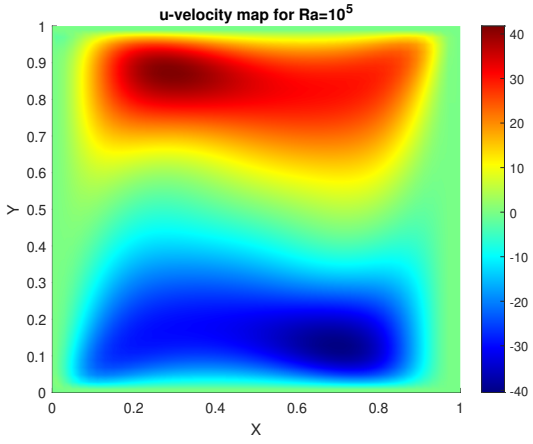
The velocity map verifies that the center of the cavity is in complete quietness. The circulation of the fluid is no longer constant. It is observed how the vertical component is taking more importance than the horizontal one. This phenomenon can be also seen in the individual velocity component maps, where the maximum vertical velocity is bigger than the horizontal one. The stagnation points of the corners have become smaller, and the center one has slightly increased.



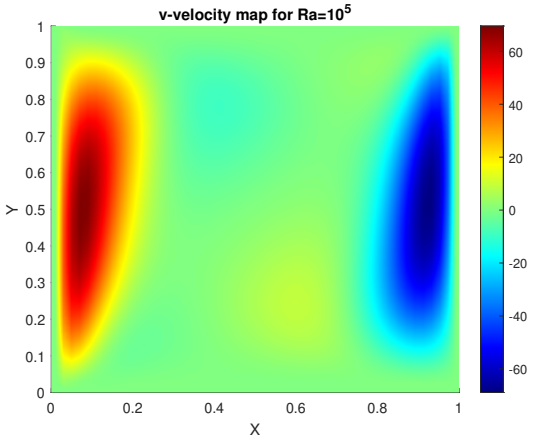
(a) Temperature map. CDS and 50×50 mesh



(b) Velocity map. CDS and 50×50 mesh



(c) Horizontal velocity map.

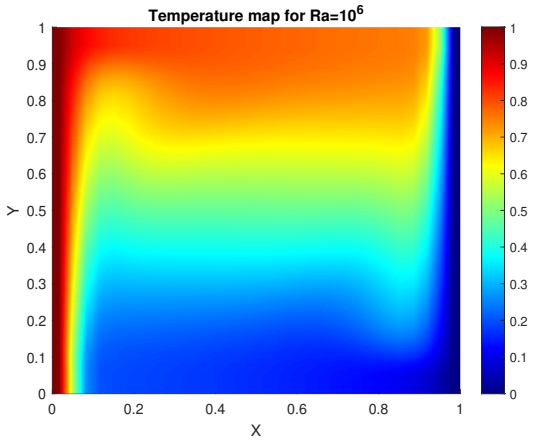


(d) Vertical velocity map.

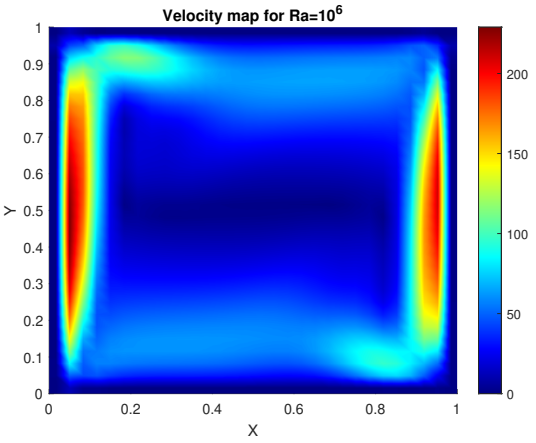
Figure 5.15: Numerical solutions for Ra=10^5

For this third simulation, an accentuation of the temperatures and velocities behavior is found. The hot fluid is taking up even more space in the upper zone, and the cold one is in the lower cavity zone. The constant temperature zone in the center has also increased in the horizontal direction.

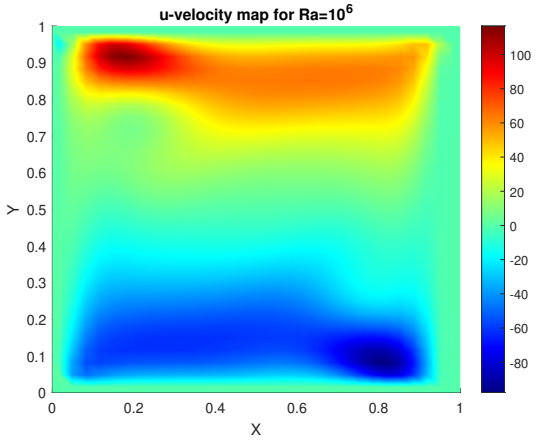
About the velocity map, the increase of the vertical component concerning the increase of the horizontal one is more significant than in the previous simulation. Even though the maximum value is increasing, the zone is getting narrower for the vertical velocity, and wider for the horizontal one. This is due to the zone occupied by the hot and cold fluids. Since more room is occupied by the hot or cold fluid, the zone of horizontal velocity is also increasing, but the zone of transition is decreasing, causing a greater gradient of vertical velocities.



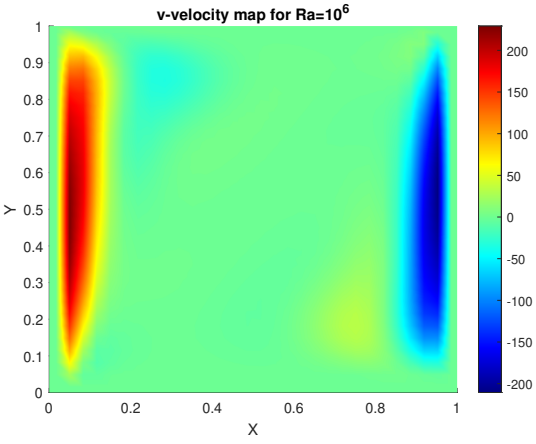
(a) Temperature map. CDS and 50×50 mesh



(b) Velocity map. CDS and 50×50 mesh



(c) Horizontal velocity map.



(d) Vertical velocity map.

Figure 5.16: Numerical solutions for Ra=10^6

This last simulation illustrates the most aggressive behavior of the fluid. The greatest temperature gradients are found near the walls and the null temperature value zone at the center of the cavity has increased up to a third of the available space.

The velocity still seems to be a representation of fluid circulation, but this time it must be noted how the vertical values are almost double the horizontal ones. For this reason, the vertical component is the dominant one, with its narrowest zone of all the previous simulations. The horizontal velocity zones, as well as the center temperature, are occupying two-thirds of the cavity, and the third left takes a null velocity value. The corner stagnation points have the smallest zones of the previous simulations, and the walls are remaining constant, due to the no-slip condition.

Table 5.2: Results comparison for Ra=10^3 and Ra=10^4. Extracted from [13]

Rayleigh	10^3	10^3	10^4	10^4
	Reference	**Numerical**	**Reference**	**Numerical**
u_{max}	3.649	3.645	16.178	15.595
y	0.813	0.810	0.823	0.81
v_{max}	3.697	3.665	19.617	17.988
x	0.178	0.190	0.119	0.11
Nu	1.118	1.079	2.243	2.078
Nu_{mid}	1.118	1.155	2.243	2.216
Nu_0	1.117	1.048	2.238	2.371
Nu_{max}	1.505	1.417	3.528	3.497
y	0.092	0.086	0.143	0.13
Nu_{min}	0.692	0.698	0.586	0.472
y	1.000	0.990	1.000	0.990

Table 5.3: Results comparison for Ra=10^5 and Ra=10^6. Extracted from [13]

Rayleigh	10^5	10^5	10^6	10^6
	Reference	**Numerical**	**Reference**	**Numerical**
u_{max}	34.73	34.253	64.63	63.354
y	0.855	0.850	0.850	0.850
v_{max}	68.59	69.535	219.36	218.54
x	0.066	0.070	0.0379	0.033
Nu	4.519	4.382	8.800	8.430
Nu_{mid}	4.519	4.383	8.799	8.427
Nu_0	4.509	4.430	8.817	8.547
Nu_{max}	7.717	7.831	17.925	18.042
y	0.081	0.090	0.0378	0.035
Nu_{min}	0.729	0.675	0.989	0.865
y	1.000	0.990	1.000	0.970

It is verified that the obtained results are very similar to the reference ones. The induced error can be due to the chosen convective scheme. As was seen in the previous benchmark problem, the Lid-Driven Cavity case, convective schemes such as SUDS are much more accurate with the solutions than CDS. In addition, the mesh size can also be an important parameter of error introduction. Finer meshes, with 100×100 of even finer ones, can reduce the error significantly, in exchange for a considerable computational cost.

5.4.3 Flow between Parallel Plates

This third benchmark problem is interesting due to its ease to be verified with an analytic solution. The problem states that there is an incompressible and laminar flow between two plane parallel plates of length L and infinite width, separated by an H distance. The flow is generated due to a known inlet velocity with a hyperbolic profile.

The boundary conditions of the problem are the following:

1. **Left side:** as said before, an inlet velocity is imposed with only a horizontal component:

$$u = u_{ref}\left(1 - \frac{4y^2}{H^2}\right) \tag{5.56}$$

2. **Right side:** this side has a Neumann boundary condition, both for horizontal velocity and for pressure.

$$\frac{\partial u}{\partial x} = 0 \tag{5.57}$$

$$\frac{\partial p}{\partial x} = 0 \tag{5.58}$$

3. **Upper and lower sides:** the no-slip condition is applied, together with a Neumann boundary condition for pressure.

$$\vec{v} = 0 \tag{5.59}$$

$$\frac{\partial p}{\partial y} = 0 \tag{5.60}$$

The following figure illustrates the geometry and boundary conditions of the problem:

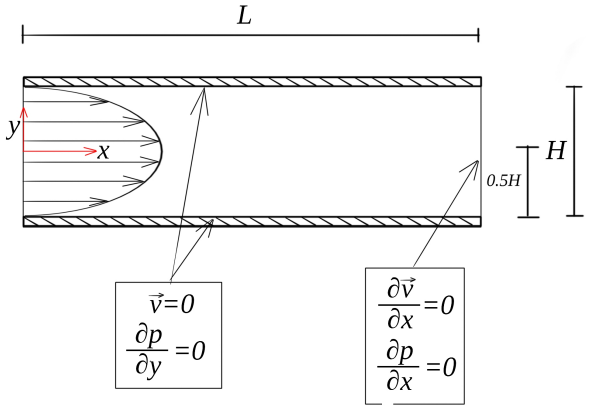


Figure 5.17: Flow between Parallel Plates scheme. Geometry and boundary conditions. Case 1

The analytical solution comes from the following procedure.

Considering the horizontal component for the Navier-Stokes mass conservation and momentum equation in

2D, steady state, and incompressible flow, the resultant equations are the following:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{5.61}$$

$$\rho\left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right) = -\frac{\partial p}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{5.62}$$

Being a steady-state flow, the equation (5.61) can be simplified to:

$$\frac{\partial v}{\partial y} = 0 \tag{5.63}$$

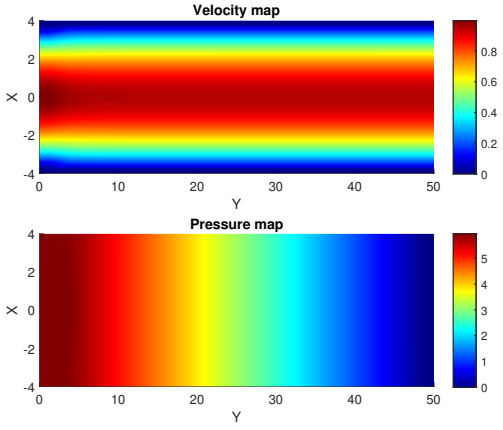
Introducing the previous equation into equation (5.62), the resultant momentum equation is written as follows:

$$\frac{\partial p}{\partial x} = \mu\frac{\partial^2 u}{\partial y^2} \tag{5.64}$$

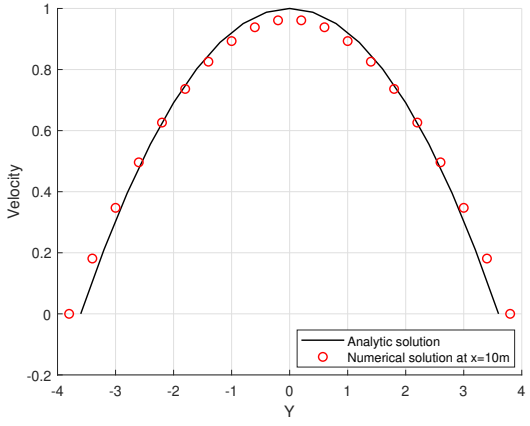
The previous equation is integrated and the horizontal velocity equation is found:

$$u(y) = \frac{1}{2\mu}\frac{\partial p}{\partial x}\left(y^2 - \frac{H^2}{4}\right) \tag{5.65}$$

The obtained results are presented below:



(a) Velocity and pressure maps for $u_{ref} = 1$



(b) Comparison between analytical and numerical solutions

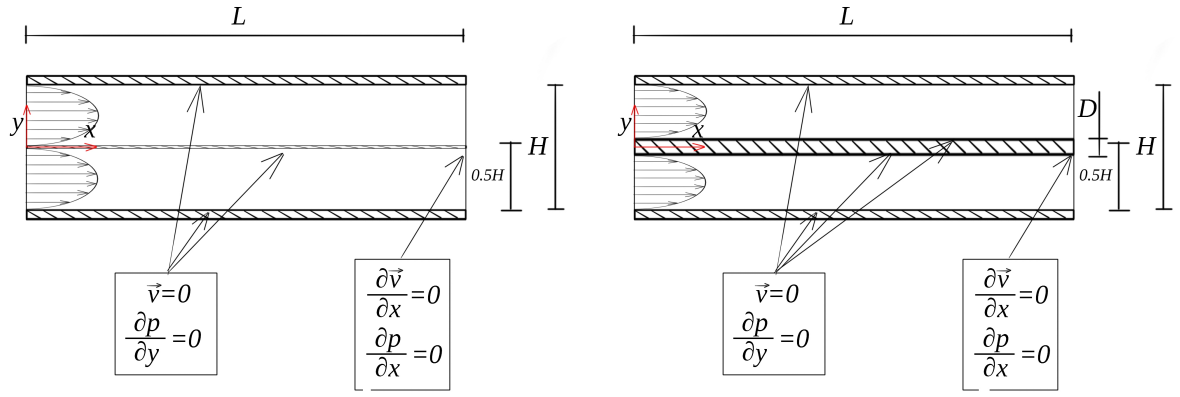
Figure 5.18: Numerical solutions for Case 1

It is visible that the numerical results almost match the analytical solution, due to spatial discretization. In addition, the pressure map shows how an inlet velocity causes a pressure gradient along the domain.

Preparing the last benchmark problem, which corresponds to a square cylinder submerged in the same very channel, some other cases are simulated, in order to verify the code and introduce the solid concept inside the flow. The first variation is to add a solid plane of infinitesimal thickness along the channel (*see Figure (5.19a)*). Taking the code a step further, the second variation considers that all the nodes of the horizontal centerline are solid, and the contiguous nodes have the no-slip condition (*see Figure (5.19b)*). The same

boundary conditions are applied and extended to the new solid plate.

As before, the following illustration shows the geometry and the boundary conditions of both cases.



(a) Case 2 (b) Case 3

Figure 5.19: Flow between parallel plates scheme. Geometry and boundary conditions. Cases 2 and 3

Translating the second case problem to a numerical scheme, it must be considered that there's no solid inside the fluid, but the two node centerlines must have null velocity due to the no-slip condition. Regarding pressure, every node will have a pressure value. and Figure (5.20) shows the numerical results obtained for this case.

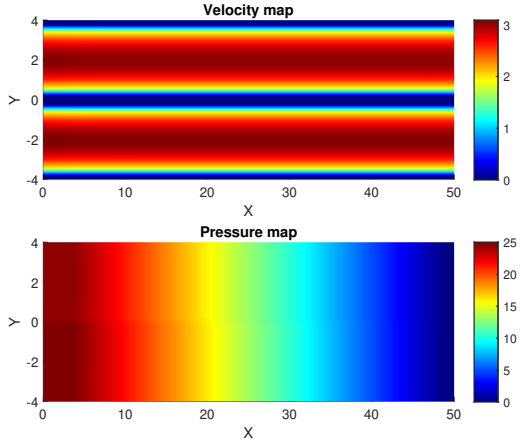
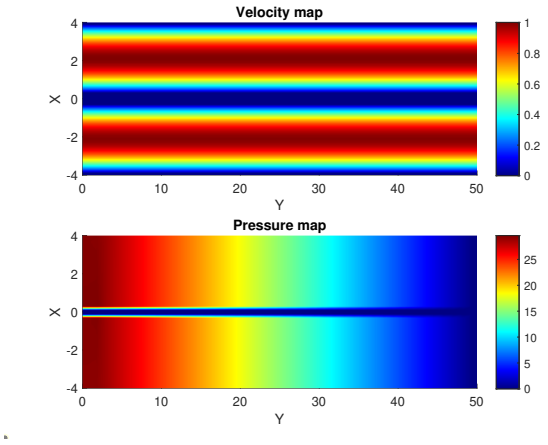
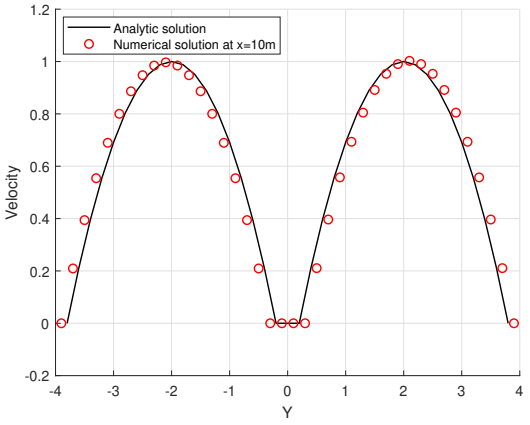


Figure 5.20: Numerical solutions for Case 2

For the third case problem, as said before there are two lines of nodes that are considered solid. Thus, the velocity of those nodes will take a null value, together with the contiguous one, and the solid nodes will have null pressure. The numerical results are the following:



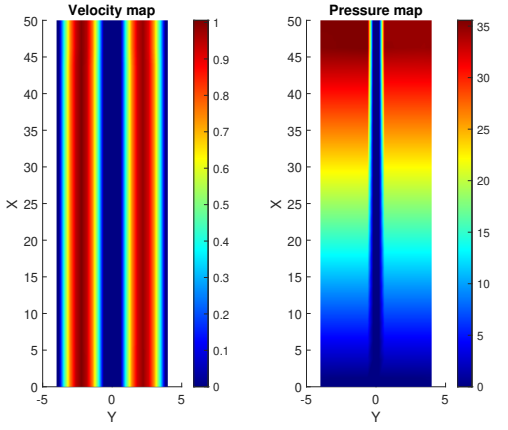
(a) Velocity and pressure maps for $u_{ref} = 1$



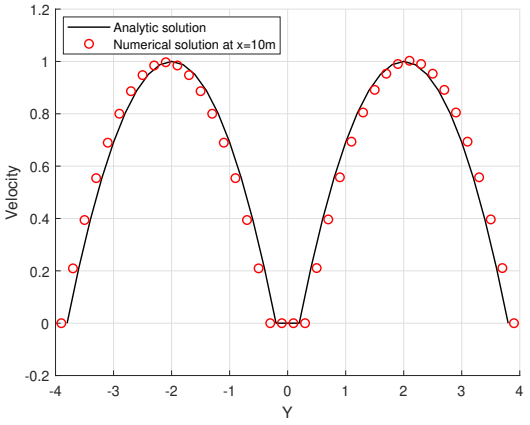
(b) Comparison between analytical and numerical solutions

Figure 5.21: Numerical solutions for Case 3

Again, the numerical results obtained match the analytical solutions. Last, but not least, an extension of the same verification case is done, considering a vertical flow instead of a horizontal one. Regarding the boundary conditions and the analytical solution, x and y components are respectively changed, and the obtained results are the following:



(a) Velocity and pressure maps for $v_{ref} = 1$



(b) Comparison between analytical and numerical solutions

Figure 5.22: Numerical solutions for Case 3 - Vertical

5.4.4 Square Cylinder

This is the last benchmark problem. As said before, is an extension of the third case of the previous benchmark problem, but this time an isolated square cylinder is considered [14].

The boundary conditions of the problem are exactly the same as before, but now extended to the solid cylinder:

1. **Left side:** hyperbolic inlet velocity is imposed with only a horizontal component:

$$u = u_{ref}\left(1 - \frac{4y^2}{H^2}\right) \tag{5.66}$$

2. **Right side:** this side has a Neumann boundary condition, both for horizontal velocity and for pressure.

$$\frac{\partial u}{\partial x} = 0 \tag{5.67}$$

$$\frac{\partial p}{\partial x} = 0 \tag{5.68}$$

3. **Upper and lower sides:** the no-slip condition is applied, together with a Neumann boundary condition for pressure.

$$\vec{v} = 0 \tag{5.69}$$

$$\frac{\partial p}{\partial y} = 0 \tag{5.70}$$

4. **Solid walls:** the same conditions are applied to the four walls of the solid. For velocity, the no-slip condition is applied, and for pressure, the Neumann boundary condition.

5. **Solid part:** all the nodes which form the solid will take a null pressure value.

The following figure illustrates the geometry and boundary conditions of the problem:

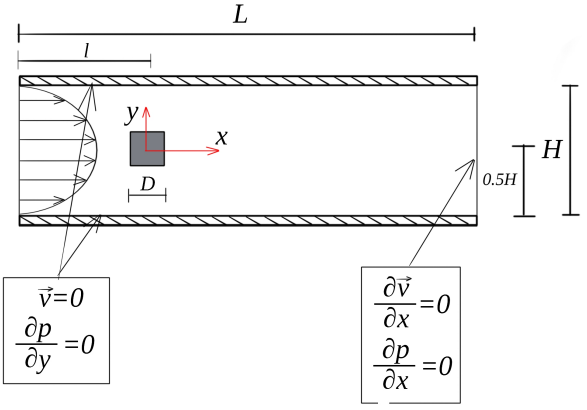


Figure 5.23: Square Cylinder scheme. Geometry and boundary conditions.

The solutions are presented below. They have been obtained through a uniform mesh of 100×50 elements in X and Y, respectively. The simulations are ruled with different Reynolds values. From $Re = 1$ to $Re = 60$, the solution is expected to be steady, but from then on, the solution becomes unsteady, so a certain time convergence criterion is applied.

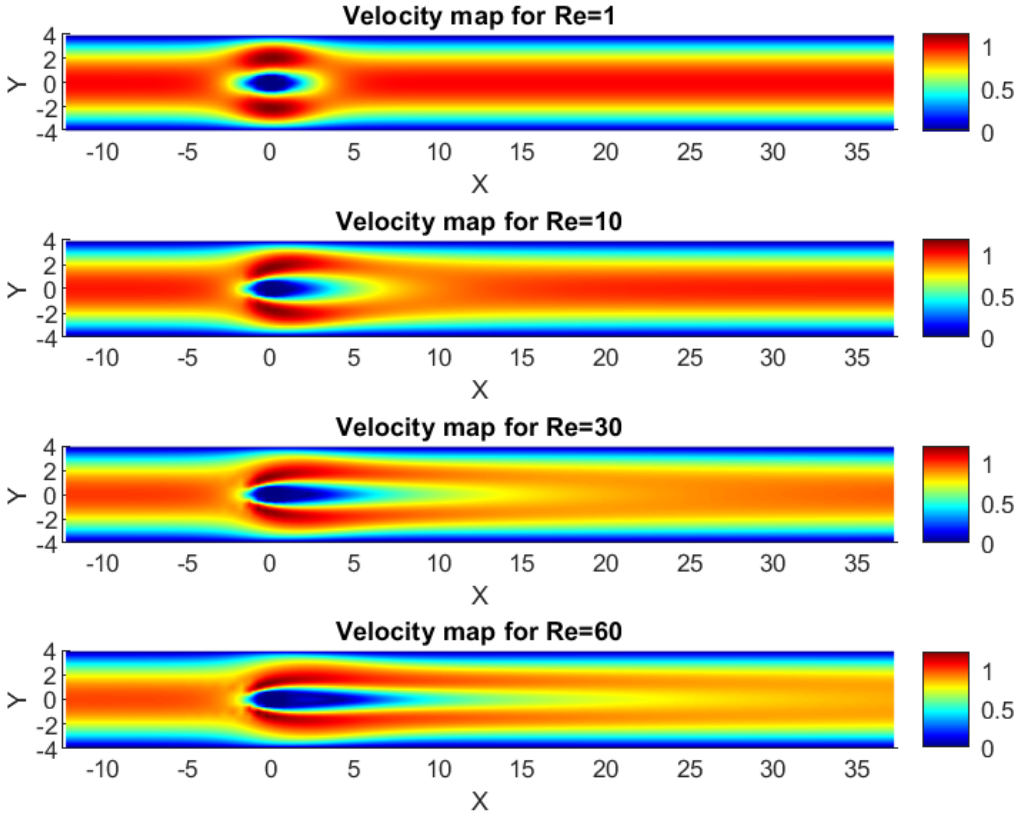


Figure 5.24: Velocity map for different Reynolds. Steady solution

The image above illustrates the numerical solution obtained for different Reynolds numbers and uniform mesh of 100×50 elements. For the lowest Reynolds value, is clearly visible how the fluid surrounds the solid and stays in the laminar regime. As the Reynolds increases, the trail of the fluid also increases its size, but in all the simulations the solution appears to be in the laminar regime.

It is important to observe and understand the recirculation zone of the solid. This zone is expected to have a recirculation of the fluid due to the separation of the boundary layer. It can be computed as follows [14]:

$$\frac{L_r}{D} = -0.065 + 0.0554Re \tag{5.71}$$

It can be verified that the obtained solution follows this behavior since the recirculation length also increases with the Reynolds number. However, a more in-depth study can not be ruled since the coarse mesh does not allow enough precise results.

Another important phenomenon that can be seen in the previous results is the stagnation point at the front of the solid. The code is capable of computing this stagnation point, where the fluid stops at the front wall of the cylinder.

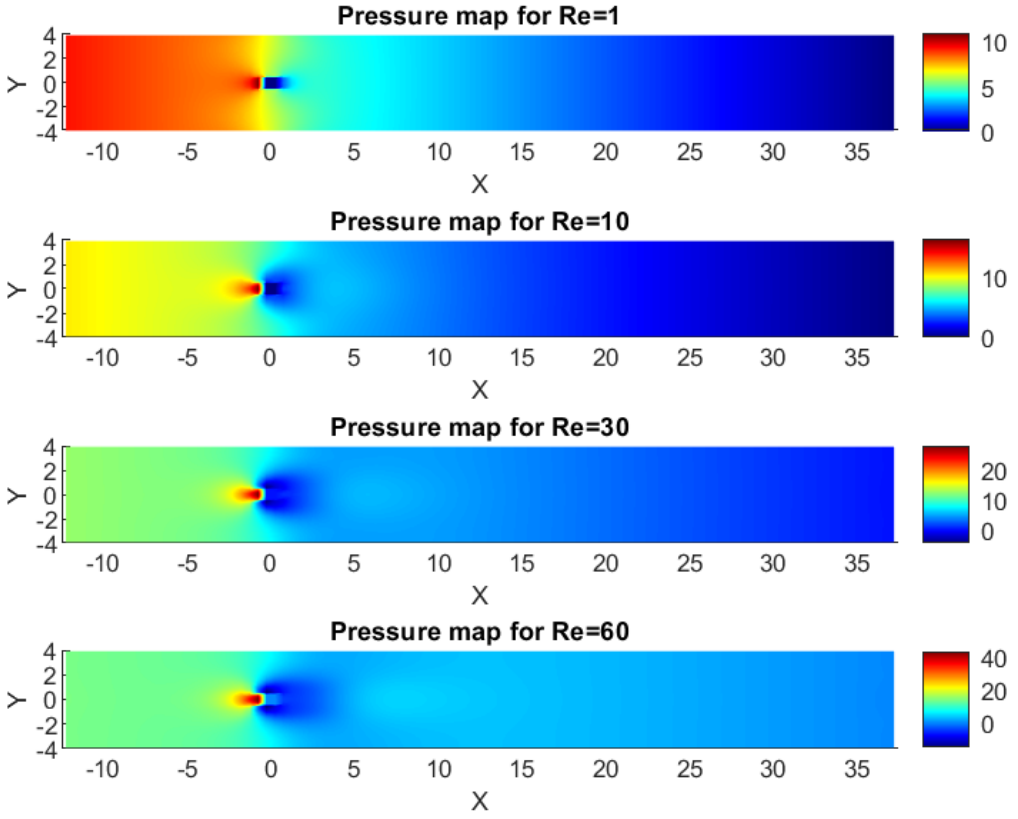


Figure 5.25: Pressure map for different Reynolds. Steady solution

Once the velocities are treated, the pressure maps can be observed here. All the simulations show a high-pressure value at the flow inlet and a low-pressure value at the outlet of the flow. Three important zones of these maps are the front wall zone or stagnation point, the solid itself, and the back wall, or recirculation zone:

1. **Stagnation point:** at this zone, it has been said that the velocity takes a null value. Consequently, the pressure map takes a high-pressure value at this zone. It can be observed how this maximum pressure value increases with the Reynolds number.

2. **Solid part:** as it was imposed in the boundary conditions, the solid nodes take a null pressure value. In the graphs, a square draft can be seen.

3. **Recirculation zone:** null or negative pressure values are expected to be found in this zone. This zone also increases with the Reynolds value. It is also known as the suction zone.

The above-mentioned results are the ones for the laminar regime. The following results are expected to have an unsteady solution, thus vortexes are going to appear in the velocity and pressure maps.

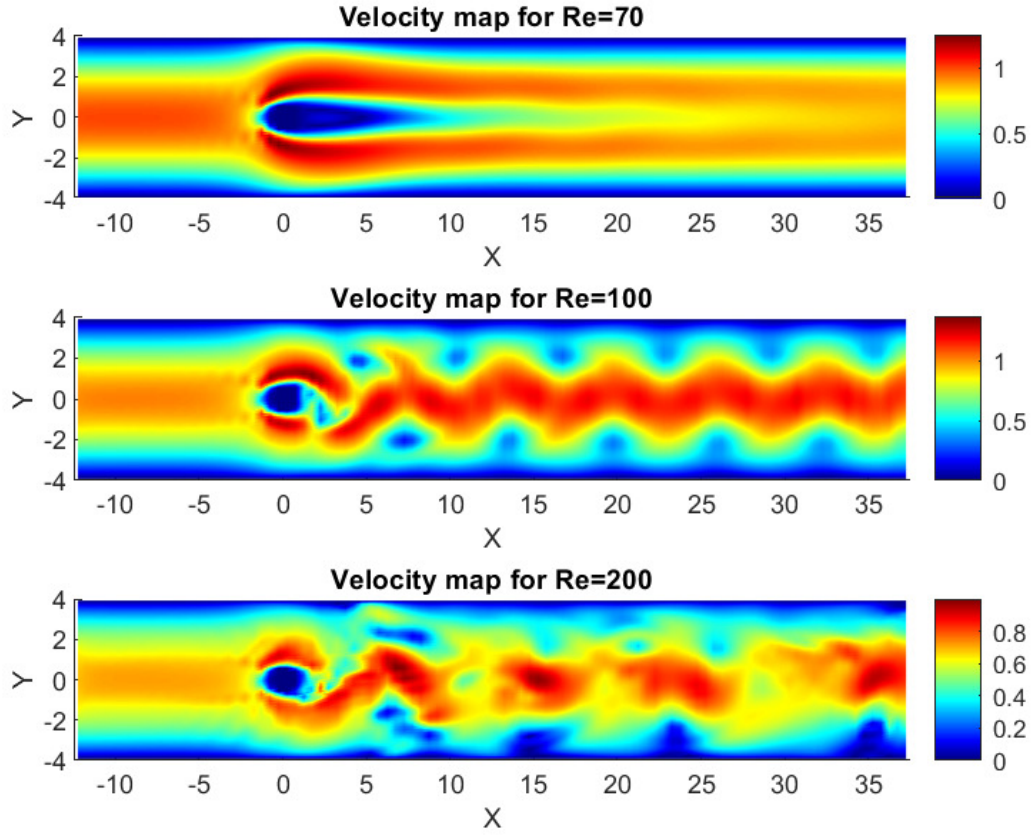


Figure 5.26: Velocity map for different Reynolds. Unsteady solution

The velocity trail for $Re = 70$ is starting to curl, which means that the critical Reynolds has been exceeded and, thus the problem becomes unsteady. As the Reynolds increases, vortexes are starting to appear.

For $Re = 100$, is clearly visible how the velocity tends to go up and down, creating the well-known Von Karman vortexes. For $Re = 200$, this behavior is much more aggressive.

It must be kept in mind that the solutions are obtained with a coarse mesh. This mesh seems to be enough precise for low Reynolds values but is visible how this mesh becomes insufficient for $Re = 200$ simulations. For this reason, the vortexes are not clearly visible nor developed.

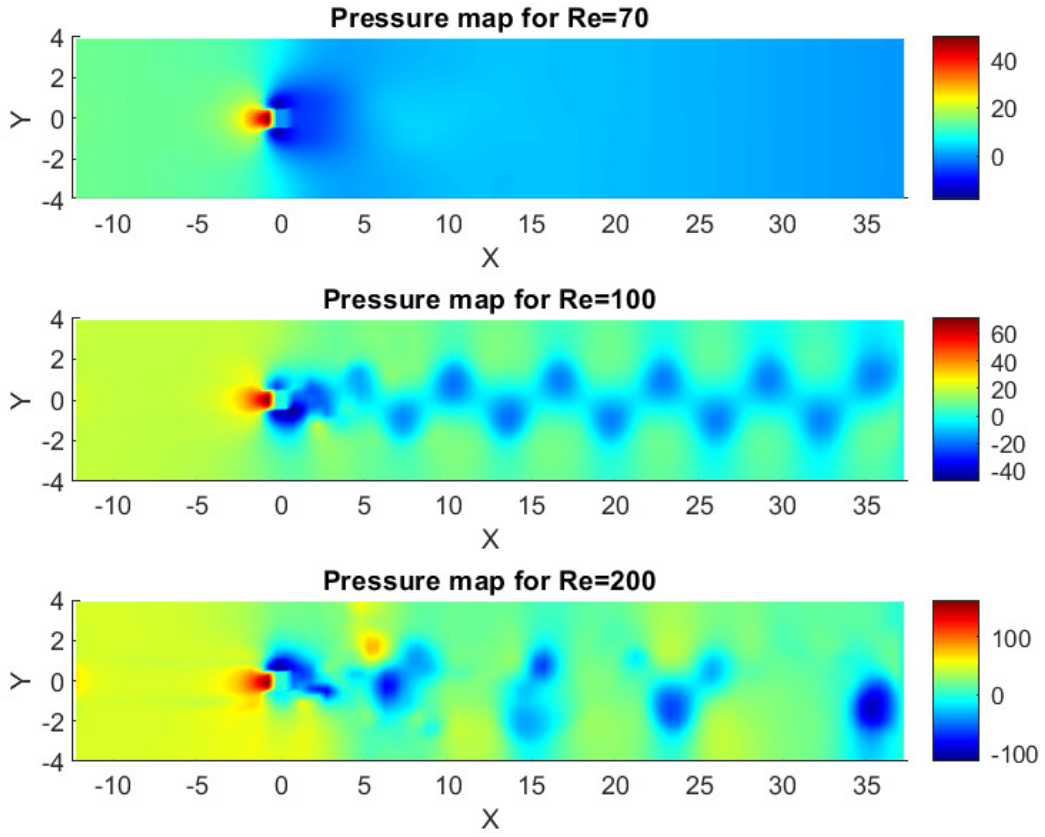
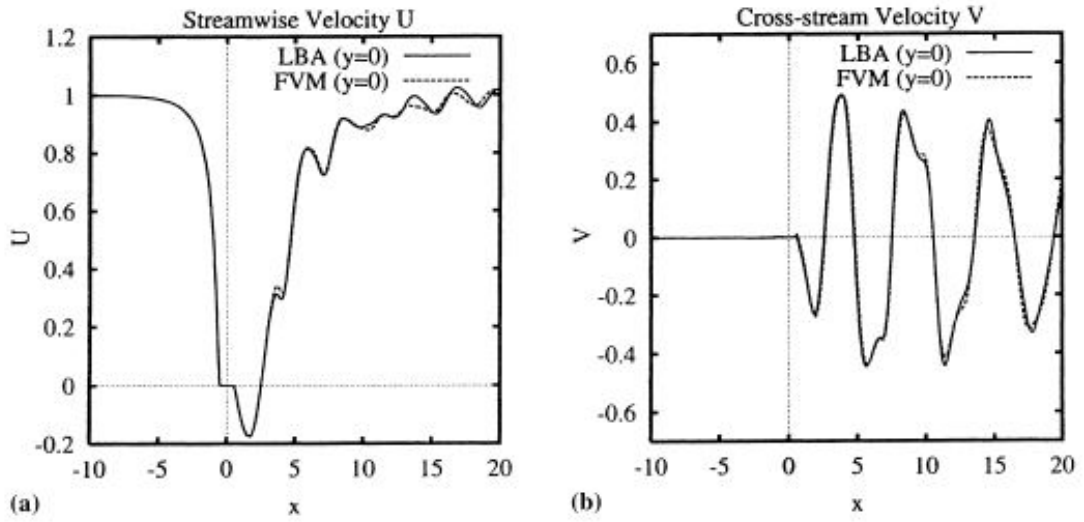


Figure 5.27: Pressure map for different Reynolds. Unsteady solution

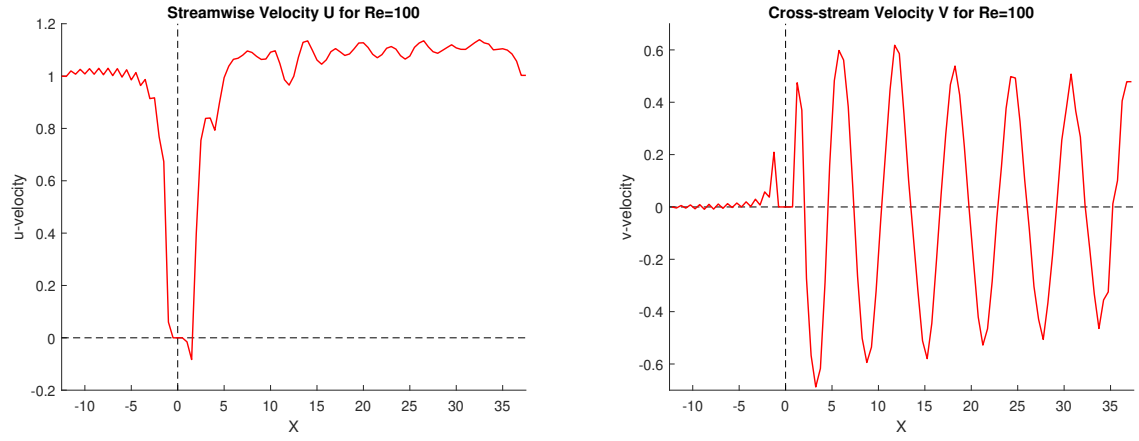
Regarding the pressure maps, for $Re = 70$ the same behavior as for the lowest Reynolds numbers is found. It is also visible how the stagnation point has a higher pressure than the previous Reynolds simulation, and the low pressure has also a lower pressure value. The solid remains at a null-pressure value.

For the other Reynolds values, the same velocity vortexes are expressed in low-pressure zones. For $Re = 100$ the solution is very clear and illustrates how the upper zone of the solid, where the flow is flowing at higher speeds, has a higher pressure value with respect to the lower zone of the cylinder. For $Re = 200$, the same problem found for the velocity map is translated into pressure values. Notwithstanding, the stagnation point and the solid itself are pretty accurate.

Finally, the last tests are ruled in order to verify the solution with respect to the reference ones. The first group of images illustrates the streamwise and cross-stream velocities at the centerline of the channel, $y = 0$ (*see Figure (5.28)*). The second group of images illustrates the same velocities at different longitudinal positions, $x = 0$, $x = 4$ and $x = 8$ (*see Figure (5.29)*).



(a) Reference results for Re=100 at $y = 0$. Extracted from [14]



(b) u-velocity map at $y = 0$

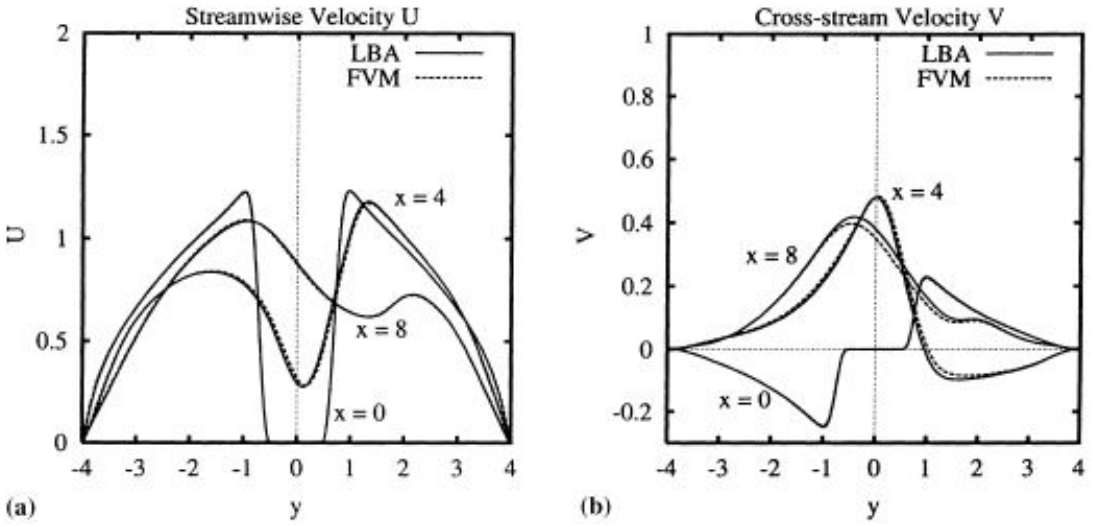
(c) v-velocity map at $y = 0$

Figure 5.28: Streamwise and cross-stream velocities for Re=100. Reference and numerical solution comparison.

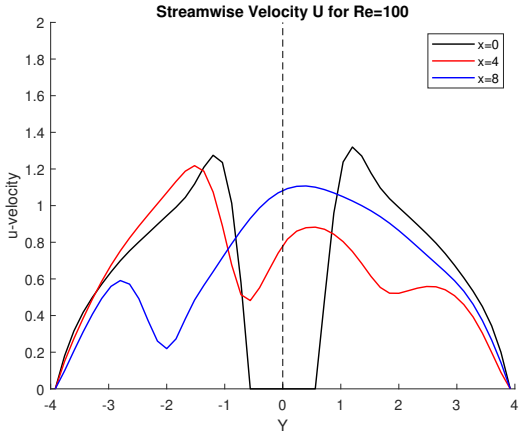
The numerical results are not perfectly matching with the reference ones for two main reasons. The first is the coarse mesh used. It has been said that for high Reynolds values, the chosen mesh is not good enough to precisely capture the flow gradients. For this reason, the solution seems to be not linear. The second reason is the time convergence criterion. The simulations are ruled up to $t = 350$s, but the reference solutions are ruled until the velocity goes from positive to a negative value at $x = 10$.

However, the streamwise representation follows the same behavior as the reference ones, with a negative drop in the velocity after the solid and a curling tendency after the solid, where the vortexes appear. The cross-stream velocity has also the same tendency due to the vortex generation.

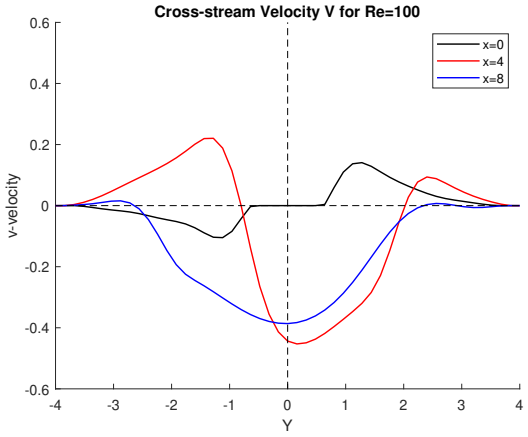
From the group of figures below, the same mesh conclusions and time convergence criterion can be extracted. It is curious to see that at the time that the simulation was stopped, the vortex was on the opposite side of the reference ones. For this reason, the Figures (5.29b) and (5.29c) are the opposite solution of the reference values at $x = 4$ and $x = 8$.



(a) Reference results for Re=100 and different x positions. Extracted from [14]



(b) u-velocity map for different x positions



(c) v-velocity map for different x positions

Figure 5.29: Streamwise and cross-stream velocities for Re=100 and different x positions. Reference and numerical solution comparison.

5.5 Discussion

Throughout the obtained results in the previous sections, it can be concluded that the algorithm presented to solve the FSM is working properly.

In the first case, the Lid-Driven Cavity, a general overview of the different convective schemes was provided, concluding that CDS is the most suitable scheme for low Reynolds. The following cases were evaluated with this scheme. In the second case, it can be also concluded that the energy equation has been successfully solved. Finally, the Flow between Plates also agrees with the analytical solution, and the Square Cylinder with the reference results.

Since all the problems have been tested with the same algorithm and the only differences between them are the geometry and the boundary conditions, it can be concluded that the code is totally capable to deal with any other problem.

Chapter 6

Turbulence

This sixth chapter is focused on the study of the turbulence phenomenon. For this purpose, a previous introduction to the phenomena' characteristics and a short justification can be found. Then, a mathematical study is done to better understand the formulation and physics behind Burgers' equation and its transformation into the Fourier space, and after that, it is applied to a theoretical case.

6.1 Introduction to Turbulence

Turbulent flow is encountered once the Reynolds number exceeds the critical Reynolds. Below this value, the flow is defined as laminar, but beyond it, the flow becomes unstable, causing chaotic and continuously changing velocity, pressure, density, and temperature maps. These fluctuations are three-dimensional and can be interpreted as turbulent eddies [15].

The energy spectrum of the turbulent system is defined by the energy distribution of the eddies in front of its wave number k. This wave number is also inversely proportional to the wavelength of the eddy. The largest length scales, also known as driving scales, are found for the lowest wave numbers, and the *Energetic Range* can be defined. The smallest length scales, also known as Kolmogorov's length scales, are found for the highest wave numbers, defining the *Dissipation Range*. Energy is transported from the largest to the smallest scales, defining the *Inertial Range*. These dependencies and ranges can be seen in Figure (6.1).

It is considered that the *Energetic Range* is an input of energy, ϵ_{in}. The *Inertial Range* is considered to be a transference of energy from the largest to the smallest scales. Is defined by a constant line with a negative slope of $-5/3$, and Kolmogorov's energy spectrum is defined by the following equation:

$$E(k) = C_k \epsilon^{2/3} k^{-5/3} \tag{6.1}$$

where C_k takes the value of 1.4. This range is affected by the Reynolds number. When Reynolds number increases, the range becomes larger because even smaller scales are developed. The last range, *Dissipation Range*, is considered to be the output of energy, which is dissipated through the smallest eddies. By the

equilibrium hypothesis, it can be assumed that $\epsilon_{out} = \epsilon_{in}$.

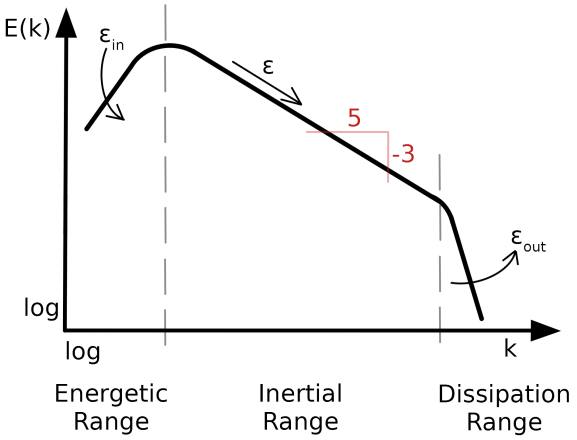


Figure 6.1: Energy cascade

It has been mentioned that the *Inertial Range* has a Reynolds dependency, so there is also a dependency between the largest and the smallest scales. Considering ϵ as an input or output of energy per unit of time, and assuming U for the largest scales and u for the smallest ones, being $U >> u$, both input and output energies are the order of:

$$\epsilon_{in} \sim \frac{U}{\frac{l}{U}} = \frac{U^3}{l} \tag{6.2}$$

$$\epsilon_{out} \sim \nu \left(\frac{u}{\eta}\right)^2 \tag{6.3}$$

And Reynolds numbers can be considered as:

$$Re_l = \frac{Ul}{\nu} \tag{6.4}$$

$$Re_\eta = \frac{U\eta}{\nu} \tag{6.5}$$

Since Reynolds number is the relation between convection and diffusion, as the energy $E(k)$ decreases, the convective term becomes weaker and the diffusive one stronger. When the energy $E(k)$ is the minimum, the smallest scale is reached, so the convective and diffusive terms have the same order of magnitude, and $Re_\eta \sim 1$.

$$U = \frac{Re_l \nu}{l} \tag{6.6}$$

$$u = \frac{Re_\eta \nu}{\eta} \sim \frac{\nu}{\eta} \tag{6.7}$$

Assuming equilibrium, $\epsilon_{in} = \epsilon_{out}$:

$$\frac{\left(\frac{Re_l \nu}{l}\right)^3}{l} \sim \nu \left(\frac{\frac{\nu}{\eta}}{\eta}\right)^2 \tag{6.8}$$

By operating the previous equation, the following relations can be found:

$$\frac{l}{\eta} \sim Re_l^{3/4} \tag{6.9}$$

$$\frac{U}{u} \sim Re_l^{1/4} \tag{6.10}$$

$$\frac{t_l}{t_\eta} \sim Re_l^{1/2} \tag{6.11}$$

Now that the relation between the largest and the smallest scales is already known, let's consider a simple three-dimensional case:

1. 3D space: $\left(\frac{l}{\eta}\right)^3 \sim Re_l^{9/4}$
2. Time: $\frac{t_l}{t_\eta} \sim Re_l^{1/2}$
3. Space × Time: $\left(\frac{l}{\eta}\right)^3 \frac{t_l}{t_\eta} \sim Re_l^{11/4}$

The computational complexity of the Navier-Stokes equations grows in the order of $Re_l^{11/4}$. This is the reason why solving a turbulent field with even the smallest scale considered and a small enough time step is highly expensive in terms of computational effort, and this is also the reason why turbulence models are used.

The simulations of turbulent fields are divided into three main parts:

1. **DNS:** Direct Numerical Simulations (DNS) are the ones that need high computational resources since they solve from the largest to the smallest length scales and the time steps are small enough to solve even the fastest fluctuations.
2. **LES:** Large Eddy Simulations (LES) are turbulent models that simplify Navier-Stokes equations and only solve the largest scales. The computational cost is reduced regarding DNS.
3. **RANS:** Reynolds Averaged Navier-Stokes Simulations (RANS) have the lowest computational cost and are obtained by averaging the fluid motion equations. This particular case is not applied in this section.

6.2 Burgers' equation in Fourier space

The Burgers' equation is a simplified model that shares many of the aspects with the Navier-Stokes equations [16]. In a one-dimensional domain, the equation in the physical space for the velocity field $u(x,t)$ is defined as:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2} + F \tag{6.12}$$

This equation is going to be solved in Fourier space on an interval Ω with periodic boundary conditions. The first step is to apply the Fast Fourier Transform (FFT) to the velocity field:

$$u(x,t) = \sum_{k=-\infty}^{k=+\infty} \hat{u}_k e^{ikx} \tag{6.13}$$

In the previous equation, k represents the wave number. Since $u(x,t) \in \mathbb{R}$, the following relation is applied:

$$\hat{u}_k = \overline{\hat{u}_{-k}} \tag{6.14}$$

Relation (6.13) is applied to each term of the equation (6.12):

1. Transient term:

$$\frac{\partial u}{\partial t} = \sum_{k=-\infty}^{k=+\infty} \frac{\partial \hat{u}_k}{\partial t} e^{ikx} \tag{6.15}$$

2. Convective term:

$$u\frac{\partial u}{\partial x} = \sum_{p=-\infty}^{p=+\infty} \hat{u}_p e^{ipx} \cdot \sum_{q=-\infty}^{q=+\infty} (iq)\hat{u}_q e^{iqx} = \sum_{p=q=-\infty}^{+\infty} \hat{u}_p (iq) \hat{u}_q e^{i(p+q)x} \tag{6.16}$$

3. Diffusive term:

$$\frac{1}{Re}\frac{\partial^2 u}{\partial x^2} = \sum_{k=-\infty}^{k=+\infty} \hat{u}_k (ik)^2 e^{ikx} \text{ and being } i^2 = -1, \text{ results in: } -\frac{k^2}{Re} \sum_{k=-\infty}^{k=+\infty} \hat{u}_k e^{ikx} \tag{6.17}$$

4. Source term:

$$F = \sum_{k=-\infty}^{k=+\infty} \hat{F}_k e^{ikx} \tag{6.18}$$

Since the summation can not be computed from $-\infty$ to $+\infty$, a discretization of N-modes is done, being the new k range $k = -N, ..., 0, ..., N$. The Burgers' equation in Fourier space stands for:

$$\sum_{k=-N}^{k=+N} \frac{\partial \hat{u}_k}{\partial t} e^{ikx} + \sum_{p=q=-N}^{+N} \hat{u}_p (iq) \hat{u}_q e^{i(p+q)x} = -\frac{k^2}{Re} \sum_{k=-N}^{k=+N} \hat{u}_k e^{ikx} + \sum_{k=-N}^{k=+N} \hat{F}_k e^{ikx} \tag{6.19}$$

The previous equation can be solved by each mode separately, considering the condition $p + q = k$ for the convective term. It can also be added that only positive modes are needed to be solved since negative ones come directly from relation (6.14). The simplified equation is the following one:

$$\frac{\partial \hat{u}_k}{\partial t} + \sum_{p+q=k} \hat{u}_p (iq) \hat{u}_q = -\frac{k^2}{Re}\hat{u}_k + \hat{F}_k \qquad \text{where} \qquad k = 0, ..., N \tag{6.20}$$

Regarding boundary conditions, the initial velocity for a generic k is defined as:

$$\hat{u}_k = \frac{1}{|k|} \tag{6.21}$$

and if the node is the null one, there is no interaction with other modes and, therefore, no mean flow is assumed, so:

$$\hat{u}_0 = 0 \tag{6.22}$$

Regarding the forcing term, $\hat{F}_k = 0$ for $k > 1$ and F_1 such that $\partial \hat{u}_1 / \partial t = 0$ for $t > 0$.

6.2.1 LES model

The implementation of a LES model is done to improve the results of the coarse meshes with no need to dense them.

In this chapter, the spectral eddy viscosity model is implemented. An additional non-dimensional viscosity multiplying the diffusive term is added, and it is determined by assuming some properties of the energy spectrum. The turbulent viscosity is defined as follows:

$$\nu_t = \nu_t^{+\infty} \left(\frac{E_{kN}}{k_N} \right)^{1/2} \nu_t^* \left(\frac{k}{k_N} \right) \tag{6.23}$$

being

$$\nu_t^{+\infty} = 0.31 \frac{5-m}{m+1} \sqrt{3-m} C_k^{-3/2} \tag{6.24}$$

$$\nu_t^* \left(\frac{k}{k_N} \right) = 1 + 34.5 e^{-3.03(k_N/k)} \tag{6.25}$$

In the previous equations, m stands for the slope of the energy spectrum, E_{kN} is the energy at the cutoff frequency, k_N, and C_k is the Kolmogorov constant. With these equations, the simplified Burgers' equation (6.20) is slightly modified:

$$\frac{\partial \hat{u}_k}{\partial t} + \sum_{p+q=k} \hat{u}_p (iq) \hat{u}_q = - \left(\frac{k^2}{Re} + \nu_t \right) \hat{u}_k + \hat{F}_k \qquad \text{where} \qquad k = 0, ..., N \tag{6.26}$$

6.2.2 Burgers' equation resolution

Before time integration of the Burgers' equation (6.20), a rearrangement is done, introducing an $R(\hat{u}_k)$ coefficient:

$$\frac{\partial \hat{u}_k}{\partial t} = R(\hat{u}_k) \quad \text{where} \quad R(\hat{u}_k) = -\frac{k^2}{Re} \hat{u}_k + \hat{F}_k - \sum_{p+q=k} \hat{u}_p (iq) \hat{u}_q \tag{6.27}$$

Time integration must be verified at any time between t^n and t^{n+1}, so it results as follows:

$$\frac{\hat{u}_k^{n+1} - \hat{u}_k^n}{\Delta t} = \frac{3}{2} R(\hat{u}_k^n) - \frac{1}{2} R(\hat{u}_k^{n-1}) \tag{6.28}$$

Due to the implicit scheme, time stability conditions must be taken into account. As in the previous chapter, the CFL condition is applied:

$$\Delta t \leq C \frac{Re}{N^2} \tag{6.29}$$

C stands for a relaxation factor between 0 and 1.

Another consideration must be taken into account regarding the convective term. Due to the Fourier transform symmetry, it has been said that positive modes are the only ones needed to be solved. In addition, contributions to the convective term that verify $p + q > N$ are also discarded because the Fourier series is truncated for $|k| \leq N$. Notice that some of these interactions involve modes with $p < 0$ or $q < 0$. In this case, negative modes are related to positive ones via equation (6.14). The following image represents all the

possible triadic interactions between modes.

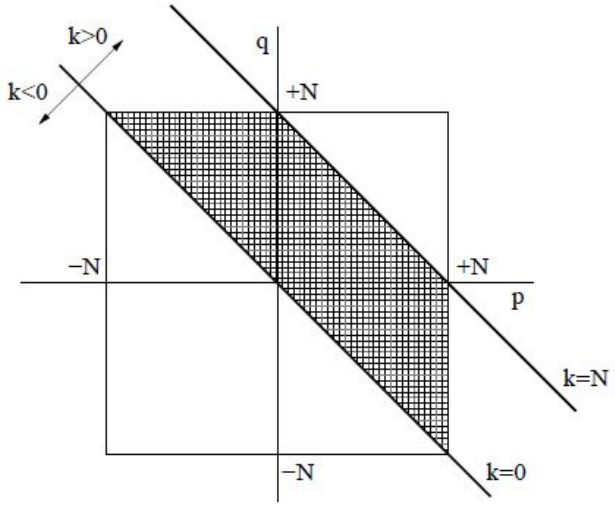


Figure 6.2: Representation of all possible triadic interactions between modes. Extracted from [17]

6.3 Code structure

The algorithm to solve the proposed problem is presented below.

1. Input data:
   (a) Physical data, such as Reynolds number and LES parameters.
   (b) Numerical data, such as convergence parameter, modes number, time step, etc.
2. CFL condition: minimum time step evaluation.
3. Initial maps for $t = 0$: $\hat{u}_k(t = 0)$
4. $R(\hat{u}_k^{n-1})$ evaluation
5. Time-convergence starting point: $t = t + \Delta t$
   (a) $R(\hat{u}_k^n)$ evaluation
   (b) Next velocity evaluation $\hat{u}_k^{n+1}$
   (c) Convergence analysis: if the steady state is reached, $|\hat{u}_k^{n+1} - \hat{u}_k^n)| < \delta$ the code goes to step 6. If not, the code updates data, $R(\hat{u}_k^{n-1}) = R(\hat{u}_k^n)$, $\hat{u}^{n-1} = \hat{u}^n$, $\hat{u}^n = \hat{u}^{n+1}$, and goes back to step 5*a*.
6. Kinetic energy computation: $E_k = \hat{u}_k \overline{\hat{u}_k}$
7. Final computations and results printing

6.4 Numerical results

6.4.1 Code verification. Re=40

The Burgers' equation is solved for $Re = 40$ for $N = 100$ (DNS) and $N = 20$, which solution is clearly underresolved. In addition, the LES model is applied for $N = 20$, $m = -2$, and $C_k = 0.4523$ or $C_k = 0.050$:

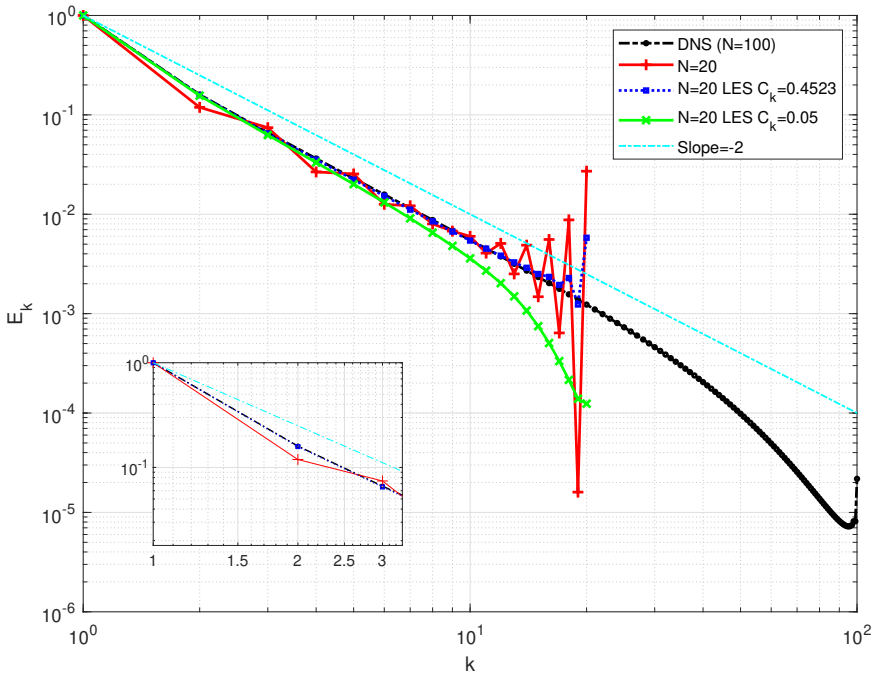


Figure 6.3: Energy spectrum of the steady-state solution of the Burgers equation for Re=40

The results can be compared with the ones provided by the CTTC:

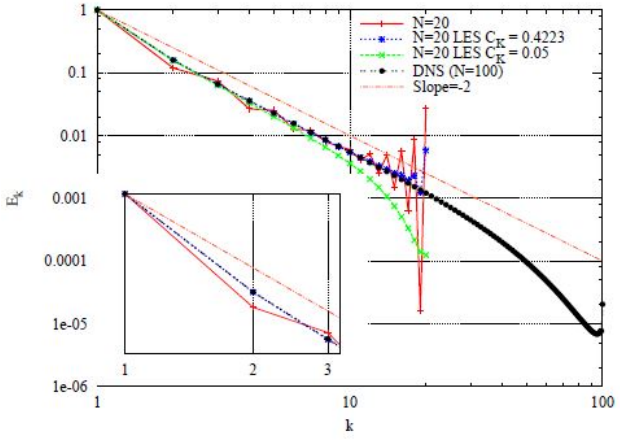


Figure 6.4: Energy spectrum of the steady-state solution of the Burgers equation. Extracted from [17]

The code can be verified since the obtained results are exactly the same provided by the CTTC report.

Regarding the DNS results, with $N = 100$, it can be seen that the energy decreases with a constant negative

slope line in the logarithmic scale until $k = 40$ approximately. This range can be identified as the *Inertial Range*. From here on, the energy decreases faster due to the increase in the dissipation of energy at the smallest scales. This last range can be identified as the *Dissipation Range*.

The simulation without a LES implementation and $N = 20$ is clearly underresolved. The results become unstable due to the lack of the smallest scales able to dissipate the energy. Here it reaffirmed the need for the small scales simulation and, therefore, the huge computational effort needed to carry out the simulations. This computational effort can be significantly reduced by the LES implementation, as the simulation shows.

Implementing a LES model and maintaining the low number of modes $N = 20$, the obtained results are significantly more stable. For $C_k = 0.4523$, the results are barely modified with respect to the DNS ones, even though the smallest scales are not considered. This is because the increase in the kinematic viscosity allows a bigger energy dissipation at larger scales. Regarding the solution for $C_k = 0.05$, it can be seen that the inertial range is significantly decreased. For the same reason as before, the decrease in the Kolmogorov constant is translated into a notable kinematic viscosity, and then, the dissipation effect is larger than the real one (DNS).

6.4.2 Extension to Re=200

In order to visualize the dependency of the *Inertial Range* with the Reynolds number, it has been increased up to $Re = 200$. With this new value, the same simulations with $N = 100$ for the DNS solution, $N = 20$, and the implementation of the LES model with $N = 20$, $m = -2$, and $C_k = 0.4523$ and $C_k = 0.05$ are ruled. The results are the following:

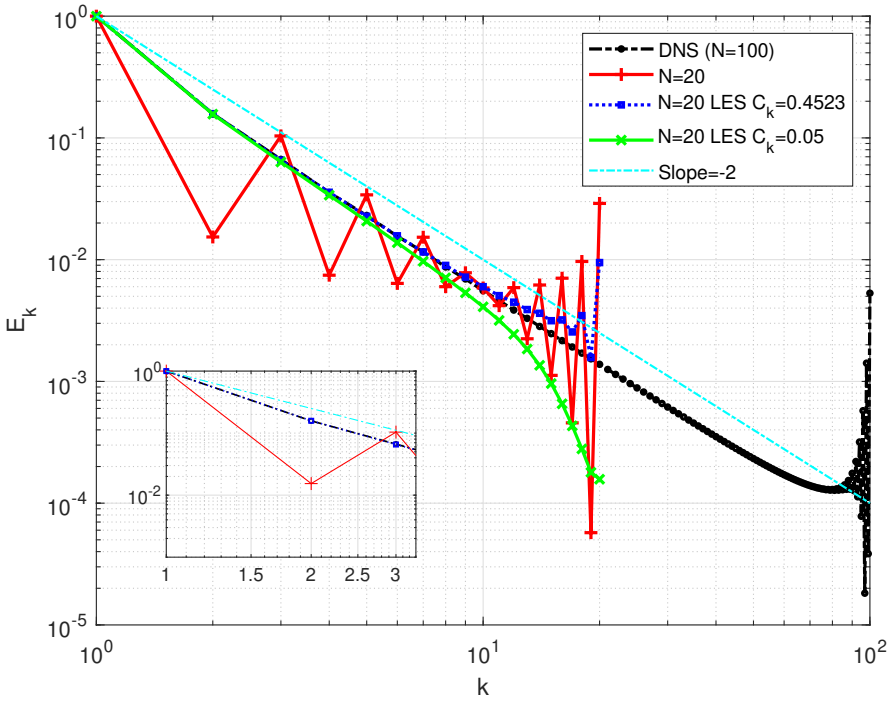


Figure 6.5: Energy spectrum of the steady-state solution of the Burgers equation for Re=200

As expected, the constant negative slope line is extended up to $k = 70$ modes for the DNS solution. From here can be also verified the code, since the *Inertial Range* has increased with the Reynolds number. It must be noted that for $k = 100$, a damping energy appears. This is the energy left for the higher modes that are not considered in the simulation. The need for the increase of the modes is logical since the Reynolds increases, and the number of modes must increase to capture the fluctuations of the turbulent flow.

Regarding the simulation with $N = 20$, the solution is highly much more unstable than the one obtained for $Re = 40$. The reason is the same as before, which is since the Reynolds has increased, much more modes are needed to capture the different fluctuations of the flow. In this case, $N = 20$ is even less valid than for the last simulation.

Chapter 7

Turbulent Flow between Parallel Plates

This seventh chapter is focused on the theoretical introduction to the study of the turbulence phenomenon applied to the flow between parallel plates problem of the previous chapter. This new case is an extension to three dimensions with two periodic directions, streamwise and spanwise. For this reason, the mathematical formulation of this chapter is only focused on the dimension extension since the FSM does not change, and the algorithm follows the same structure.

7.1 Problem definition

The main goal is the numerical resolution of a turbulent flow enclosed between ideally infinite parallel plates. Due to the lack of time before the final deadline, the problem is only studied from a theoretical point of view.

The problem states that there is an incompressible flow between two ideally infinite parallel plates [18]. Since it is desired to obtain a fully developed turbulent flow, some considerations must be done beforehand.

1. This problem operates with the friction Reynolds Re_τ. It is defined by the equation below. It is important to note that it depends on the half-width of the channel, denoted by δ_w, and also depends on the shear velocity u_τ, which takes unitary value:

$$Re_\tau = \frac{\delta_w u_\tau}{\nu} \tag{7.1}$$

   The value of the shear velocity and the half-width, and also the value of the friction Reynolds number are imposed. Therefore, the value of the kinematic viscosity is computed.

2. The geometry of the problem relies on the half-width of the channel, but also the friction Reynolds number (*see Table (7.1)*).

3. To obtain the ideally infinite parallel plates, the concept of periodic inlet conditions must be introduced.

Since it is not affordable to simulate infinite lengths from the computational point of view, it is declared that the outlet values of the properties from the previous time step are the inlet values for the next time step. This condition is applied in the streamwise and spanwise directions.

4. Due to the previous condition of the periodic inlet conditions, it can not be defined an inlet velocity. For this reason, a force term is added to the horizontal momentum equation.

The following table sums up the geometry of the problem, depending on the value of the Reynolds:

Table 7.1: Geometry of the problem. Extracted from [19]

Re_τ	**L**	**H**	**W**
180	$4\pi\delta_w$	$4/3\pi\delta_w$	$2\delta_w$
395	$2\pi\delta_w$	$\pi\delta_w$	$2\delta_w$
590	$2\pi\delta_w$	$\pi\delta_w$	$2\delta_w$

As said before, inlet conditions for streamwise and spanwise directions are declared as periodic inlet conditions. The rest of the problem boundary conditions are the following:

1. **Left side:** periodic inlet velocity:

$$\vec{v}^{n+1}[x=0] = \vec{v}^{n}[x=L] \tag{7.2}$$

2. **Right side:** this side has a Neumann boundary condition for velocity, and pressure is imposed to take a null value.

$$\frac{\partial \vec{v}}{\partial x} = 0 \tag{7.3}$$

$$p = 0 \tag{7.4}$$

3. **Upper and lower sides:** the no-slip condition is applied, together with a Neumann boundary condition for pressure.

$$\vec{v} = 0 \tag{7.5}$$

$$\frac{\partial p}{\partial y} = 0 \tag{7.6}$$

4. **Front side:** periodic inlet velocity:

$$\vec{v}^{n+1}[k=0] = \vec{v}^{n}[k=W] \tag{7.7}$$

5. **Back side:** this side has a Neumann boundary condition for horizontal:

$$\frac{\partial \vec{v}}{\partial x} = 0 \tag{7.8}$$

The following figure illustrates the geometry and boundary conditions of the problem:

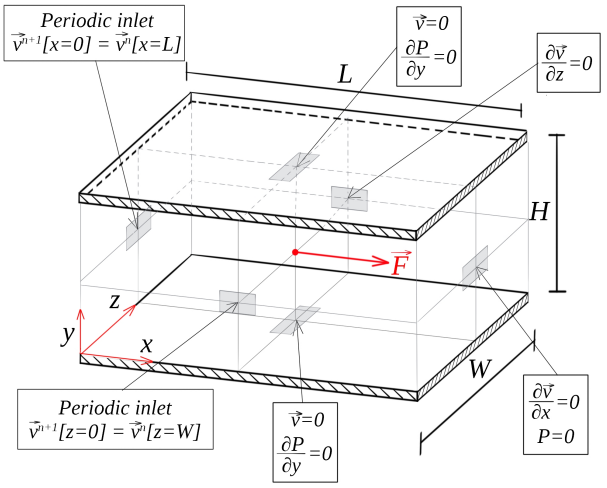


Figure 7.1: Turbulent Flow between Parallel Plates scheme. Geometry and boundary conditions.

7.2 3D extension

The extension of the FSM algorithm to three dimensions is quite easy. The following list sums up all the changes and additions that must be done to the code

1. The $R(\vec{v}^n)$ computation is the same, adding the third dimension components to the $R(u^n)$ and $R(v^n)$ coefficients, and creating the new $R(w^n)$ coefficient:

   (a) **x-staggered mesh:** apart from the new dimension coefficients, this $R(u^n)$ must include the force term that will generate the flow motion:

   $$R(u^n)\Omega_{xP} = -conv + diff + F \tag{7.9}$$

   where convective and diffusive terms are computed as it was done in the FSM code, but adding the third dimension coefficients. Note that f and b stand for *front* and *back*:

   $$conv = \dot{m}_e u_e - \dot{m}_w u_w + \dot{m}_n u_n - \dot{m}_s u_s + \dot{m}_b u_b - \dot{m}_f u_f \tag{7.10}$$

   $$diff = \mu_e \frac{u_E - u_P}{d_{PE}} A_e - \mu_w \frac{u_P - u_W}{d_{PW}} A_w + \mu_n \frac{u_N - u_P}{d_{PN}} A_n - \mu_s \frac{u_P - u_S}{d_{PS}} A_s + \\ + \mu_b \frac{u_B - u_P}{d_{PB}} A_b - \mu_f \frac{u_P - u_F}{d_{PF}} A_f \tag{7.11}$$

   Volumetric flow and velocities at the faces are computed the same way as before.

   (b) **y-staggered mesh:** the same addition of the third dimension is done. Convective and diffusive terms are the same as before but with vertical velocities. The evaluation of the velocities is done the same way as in the FSM code.

(c) **z-staggered mesh:** analogous to the previous $R(u^n)$ and $R(v^n)$, an $R(w^n)$ coefficient is computed:

$$R(w^n)\Omega_{zP} = -conv + diff \tag{7.12}$$

This time, convective and diffusive terms are evaluated with the spanwise component of the velocity, w.

2. Once $R(u^n)$, $R(v^n)$, and $R(w^n)$ are computed with the x-staggered, y-staggered, and z-staggered meshes, the predictory velocity can be computed node-by-node. u^P and v^P are computed the same way as before (*see equations (5.24) and (5.25)*), and w^P follows the same scheme:

$$w^P = w^n + \frac{\Delta t}{\rho}\left(\frac{3}{2}R(w^n) - \frac{1}{2}R(w^{n-1})\right) \tag{7.13}$$

3. The pressure is computed the same way as it was in equation (5.26), and also adding the new components due to the dimension extension:

$$a_P p_P^{n+1} = a_E p_E^{n+1} + a_W p_W^{n+1} + a_N p_N^{n+1} + a_S p_S^{n+1} + a_B p_B^{n+1} + a_F p_F^{n+1} + b_P \tag{7.14}$$

where

$$a_E = \frac{A_e}{d_{PE}} \qquad a_W = \frac{A_w}{d_{PW}} \qquad a_N = \frac{A_n}{d_{PN}} \qquad a_S = \frac{A_s}{d_{PS}} \qquad a_B = \frac{A_b}{d_{PB}} \qquad a_F = \frac{A_f}{d_{PF}} \tag{7.15}$$

$$a_P = a_E + a_W + a_N + a_S + a_B + a_F \tag{7.16}$$

$$b_P = \frac{-1}{\Delta t}\left((\rho u^P)_e A_e - (\rho u^P)_w A_w + (\rho v^P)_n A_n - (\rho v^P)_s A_s + (\rho w^P)_f A_f - (\rho w^P)_b A_b\right) \tag{7.17}$$

4. Finally, the next velocity $\vec{v}^{n+1}$ can be computed node-by-node. Once again, the horizontal and vertical components don't change from the previous expression (*see equations (5.30) and (5.31)*). The new spanwise next velocity is computed following the same scheme:

$$w_P^{n+1} = w_P^P - \frac{\Delta t}{\rho}\left(\frac{p_B^{n+1} - p_A^{n+1}}{d_{BA}}\right) \tag{7.18}$$

7.3 Future work

As said before, due to the lack of time before the final deadline, this case has not been finished. Notwithstanding, the list down below sums up the scheme that must be followed to achieve the solution:

1. The first could be the introduction of the streamwise periodic condition to the 2D code in the laminar regime. The same code used to solve the flow between parallel plates could be modified, and the obtained results must be the same.

2. Once the streamwise periodic condition is introduced, the 3D extension can be done including the steps

abovementioned. Then, the spanwise periodic condition has to be introduced also.

3. Finally, the solution for Re_τ=180 and others can be found. It must be kept in mind that the solution is found in the turbulent regime. For this reason, the time convergence criterion can no longer be the same. Instead, a certain time achievement must be imposed.

Chapter 8

Budget summary

The total budget of the thesis can be split into three parts.

The first part is the equipment and energy resources costs, which include the price of the computer and the energy consumption. The second part is the working hours' costs, which take into account the 600 hours of work applied to this thesis and the salary of a technical engineer. The third part includes the software licenses' costs. Since all the software used for this thesis is free, no costs are computed in this part.

The following table sums up the general costs of each part mentioned previously. A detailed overview of each one, along with its own cost table can be found in the budget report.

Table 8.1: Total budget of the thesis

Concept	**Total cost**
Equipment and Energy resources costs	87.43€
Working hours costs	8457.90€
Software costs	0€
TOTAL	**8545.33€**

Chapter 9

Analysis and assessment of environmental and social implications

The present chapter focuses on the analysis and assessment of the environmental impact associated with this CFD thesis. The objective is to evaluate the environmental implications resulting from the usage of a computer system and the energy source required to power the computer's battery throughout the thesis development.

To conduct this analysis, various factors related to energy consumption and environmental impact will be considered. These factors include:

1. **Energy Consumption:** the amount of energy consumed by the computer system during the thesis research can be quantified. An estimated value of 45.6W of power is consumed by the computer, and 600 hours of work have been needed to develop this thesis.
2. **Energy Source:** based on the hired energy service of the house, 27% comes from renewable sources, 24.6% comes from natural gas, 4.9% from coal, 29.5% comes from nuclear sources, and 14% from other nonrenewable sources.
3. **Greenhouse Gas Emissions:** also based on the hired energy service of the house, 0.2kg/kW·h of $C0_2$ are emitted. Taking into account the energy consumption, a total of 5.47kg of $C0_2$ are emitted.

It is important to note that this analysis is specific to the energy consumption of the computer system used in the thesis and does not encompass other potential environmental impacts such as the manufacture of the computer.

Chapter 10

Conclusions

A brief overview of the obtained results is presented down below, along with some future actions that could be carried out if an expansion of the present thesis is desired.

10.1 Conclusions

The aim of the project was the development of a set of codes capable of solving heat and mass transfer problems involving fluid dynamics. Using the $C++$ language, these codes have been programmed, verified, and validated.

The first approach to the subject was the heat conduction phenomena. After that, an extension to the general Convection-Diffusion equation was ruled. Therefore, the FSM was addressed with well-known benchmark problems and, finally, an introduction to the turbulence field was ruled by studying the Burgers' equation. Every chapter included a mathematical formulation and code structure and different verification cases which allowed the validation of the code.

At this point, it could be affirmed that the main objective of the thesis has successfully been satisfied since every code developed provided excellent results compared to the reference ones.

Some general conclusions can be drawn from the previous studies. The different theoretical demonstrations provided a general overview of the Navier-Stokes equations along with their particular hypothesis to simplify the cases. In addition, an in-depth knowledge acquisition about numerical schemes has been covered, along with spatial discretization and mesh types, and convective schemes and their main aspects. The importance of the mesh selection along with the convective scheme has been reflected throughout the whole verification process of every case. This thesis also provided the opportunity to improve the programming skills in every sense, from the simplest variable type and declaration up to the importance of code optimization, vector definitions, and the use of functions. Notwithstanding, the code debug process was slower than desired, thus final Turbulent Flow case was not finished.

10.2 Future actions

There are several aspects that, due to the duration of the work, have not been able to be addressed or studied with sufficient depth. Hereafter, a brief list with future actions is presented if the present thesis wants to be improved and expanded:

1. Programming skills have been a crucial factor in this thesis. For this reason, some formation in the field of $C++$ language would be a great help.
2. From the theoretical point of view, the Navier-Stokes equations study could be continued in order to expand the thesis to new interesting cases.
3. Regarding the numerical schemes, other solvers such as TDMA or others could be implemented to compare the solutions and performance of the codes.
4. This thesis introduced the turbulent phenomenon. It could be very interesting to continue this line. Firstly, finishing the Turbulent Flow code and obtaining the solutions and, then, implementing turbulent models such as LES or RANS to compare the obtained results.

Bibliography

1. VERSTEEG, H. K.; MALALASEKERA, W. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Education Limited, 2007. ISBN 978 0131274983.
2. CTTC. *Heat and Mass Transfer Technological Center*. 2023-06-06. Available also from: https://www.cttc.upc.edu/.
3. BLAZEK, J. *Computational Fluid Dynamics: Principles and Applications*. Elsevier Science, 2001. ISBN 978 0080432093.
4. LEVEQUE, R. J. *Finite-Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2004. ISBN 978 0521810876.
5. CHAPRA, S.C.; CANALE, R. P. *Numerical Methods for Engineers*. McGraw-Hill Education, 2015. ISBN 978 0073397924.
6. ÇENGEL, Y.A.; A.J.GHAJAR. *Heat and Mass Transfer: Fundamentals & Applications*. McGraw-Hill Education, 2015. ISBN 978 0073398181.
7. CTTC. *Two-dimensional Transient Heat Conduction Problem (four materials)*. Tech. rep. Universitat Politècnica de Catalunya.
8. CTTC. *Numerical resolution of the generic convection-diffusion equation*. Tech. rep. Universitat Politècnica de Catalunya.
9. PATANKAR, S. V. *Numerical Heat Transfer and Fluid Flow*. CRC Press, 2009. ISBN 978 0891165224.
10. CTTC. *Fractional Step Method: Staggered and Collocated Meshes*. Tech. rep. Universitat Politècnica de Catalunya.
11. GHIA, U.; GHIA, K.N.; C.T.SHIN. High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method. *Journal of Computational Physics*. 1982, vol. 48, pp. 387–411.
12. VAHLDAVIS, G De; I.P.JONES. Natural Convection in a Square Cavity: A comparison exercise. *International Journal for Numerical Methods in Fluids*. 1983, vol. 3, pp. 227–248.
13. VAHLDAVIS, G De. Natural Convection of Air in a Square Cavity: A bench mark numerical solution. *International Journal for Numerical Methods in Fluids*. 1983, vol. 3, pp. 249–264.
14. M. BREUER, et al. Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-Boltzmann and finite-volume. *International Journal of Heat and Fluid Flow*. 2000, vol. 21, pp. 186–196.
15. POPE, S. B. *Turbulent Flows*. Cambridge University Press, 2000. ISBN 978 0521598866.
16. CTTC. *Burgulence: a Good Toy Model for Turbulence*. Tech. rep. Universitat Politècnica de Catalunya.

17. CTTC. *Burguers equation in Fourier space*. 2008. Tech. rep. Universitat Politècnica de Catalunya.

18. MOSER, R.D.; KIM, J.; P.MOIN. Turbulence statistics in fully developed channel flow at low Reynolds number. *Journal of Fluid Mechanics*. 1987, vol. 177, pp. 133–166.

19. KIM, R.D. Moserand J.; N.N.MANSOUR. Direct numerical simulation of turbulent channel flow up to Re_τ=590. *Physics of Fluids*. 1999, vol. 11, no. 4, pp. 943–945.