



Title of the project:

# Study for the numerical resolution of conservation equation of mass, momentum and energy to be applied on the aerodynamics of bluff bodies:



Author: Oriol Crespo Pradell

Director: Carlos-David Pérez Segarra

Codirector: Xavier Trias Miquel

Tutor: Daniel Garcia Almiñana

Escola Tècnica Superior d'Enginyeries Industrial i Aeronàutica de Terrassa

Course: Grau en Enginyeria en Tecnologies Aeroespacials

Universitat Politècnica de Catalunya BarcelonaTech

Barcelona, 29/05/2015, convocatòria Juny

Content: Document 1- Memory of the project



Escola Tècnica Superior d'Enginyeries  
Industrial i Aeronàutica de Terrassa

UNIVERSITAT POLITÈCNICA DE CATALUNYA



## Table of contents

Acknowledgements.....	1
B1 Approach of the project.....	2
Aim of the project .....	2
Scope of the project.....	2
Basic requirements .....	3
Justification of the utility of the project .....	3
B2 Previous studies and state of the art.....	5
Previous studies .....	5
State of the art .....	5
B3: Possible solutions and justification of the proposed solution.....	9
B4 Development of the solution and justification of the results.....	10
Conduction problem .....	10
Introduction .....	10
Application of the conduction laws: .....	11
Problem solution:.....	13
Results:.....	21
The Smith Hutton problem .....	27
Introduction .....	27
Application of the convection diffusion laws:.....	28
Problem solution.....	29
Results.....	37
The driven cavity problem .....	42
Introduction .....	42
Numerical solution.....	47
Results.....	53
Reynolds 10000.....	59
Code verifications .....	60
Burgers problem .....	62



Introduction .....	62
Problem solution.....	66
Results.....	69
Flux around a square cylinder.....	73
Introduction .....	73
Numerical solution.....	75
Results.....	76
B4' Scope and specifications performance:.....	84
B5 Budget and environmental impact .....	85
Budget.....	85
Environmental impact.....	85
B6 Temporal aspects.....	87
Temporal planning .....	87
B7 Conclusions and recommendations.....	88
Conclusions .....	88
Recommendations .....	88
Bibliography .....	90

## Figure captions

Figure 1: Physical problem.....	11
Figure 2: differential control volume .....	11
Figure 3: First element division.....	13
Figure 4: Second element division .....	13
Figure 5: generic node description .....	13
Figure 6: Boundary node description.....	15
Figure 7: Temperature distribution at time=5000s .....	20
Figure 8 Node interpolation scheme .....	20
Figure 9: Temperature distribution at time=500s (first phase) .....	21
Figure 10: Temperature distribution at time=3000s (transient state) .....	22
Figure 11: Temperature distribution at time=4000s (second phase) .....	22
Figure 12: Temperature Distribution at time=10000s.....	23
Figure 13:Contour plot of the solution at time=10000s .....	24
Figure 14: Effect of the time step in the computational time .....	24
Figure 15:Effect of the time step in the solution of node A .....	24
Figure 16: Effect of the time step in the solution of node B.....	24
Figure 17: Computational time trend lines.....	25
Figure 18: Effect of the number of nodes on the solution of node A.....	26
Figure 19: Effect of the number of nodes on the solution of node B .....	26
Figure 20: Smith Hutton problem definition.....	27
Figure 21: Boundary nodes .....	32
Figure 22: Boundary properties .....	33
Figure 23: Temperature distribution for $\Gamma\rho = 10$ .....	34
Figure 24: Temperature distribution for $\Gamma\rho = 1000$ .....	35
Figure 25: Hyperbolic tangent representation .....	37
Figure 26: Temperature distribution for $\Gamma\rho = 1000$ .....	38
Figure 27: Contour plot of the temperature distribution for $\Gamma\rho = 1000$ .....	38
Figure 28: Contour plot of the temperature distribution for $\Gamma\rho = 106$ .....	39
Figure 29: Temperature distribution for $\Gamma\rho = 10$ .....	39
Figure 30: Contour plot of the temperature distribution for $\Gamma\rho = 10$ .....	40
Figure 31: Temperature at outlet for $\rho/\Gamma = 10$ .....	40
Figure 32: Temperature at outlet for $\rho/\Gamma = 103$ .....	40
Figure 33: Temperature at outlet for $\rho/\Gamma = 106$ .....	41
Figure 34: Driven cavity problem.....	42
Figure 35: Convective + viscous vector field decomposition.....	43
Figure 36: Face centred and staggered meshes .....	47
Figure 37: x and y velocity staggered meshes .....	48

Figure 38: Comparison of the horizontal speed solution for different mesh densities and the CTTC solution .....	53
Figure 39: Comparison of the vertical speed solution for different mesh densities and the CTTC solution .....	53
Figure 40: Velocity field .....	54
Figure 41: Streamline distribution .....	54
Figure 42: Comparison of the horizontal speed solution for different mesh densities and the CTTC solution .....	55
Figure 43: Comparison of the vertical speed solution for different mesh densities and the CTTC solution .....	55
Figure 44: Streamline distribution .....	55
Figure 45: Comparison of the horizontal speed solution for different mesh densities and the CTTC solution .....	56
Figure 46: Comparison of the vertical speed solution for different mesh densities and the CTTC solution .....	56
Figure 47: Streamline distribution .....	56
Figure 48: Comparison of the horizontal speed solution for different mesh densities and the CTTC solution .....	57
Figure 49: Comparison of the vertical speed solution for different mesh densities and the CTTC solution .....	57
Figure 50: Streamline distribution .....	57
Figure 51: Comparison of the horizontal speed solution for different mesh densities and the CTTC solution .....	58
Figure 52: Comparison of the vertical speed solution for different mesh densities and the CTTC solution .....	58
Figure 53: Streamline distribution .....	58
Figure 54: Comparison of the horizontal speed solution for different mesh densities and the CTTC solution .....	59
Figure 55: Comparison of the vertical speed solution for different mesh densities and the CTTC solution .....	59
Figure 56: Streamline distribution .....	59
Figure 57: Comparison of the horizontal speed solution for different mesh densities and the CTTC solution .....	60
Figure 58: Comparison of the vertical speed solution for different mesh densities and the CTTC solution .....	60
Figure 59: Streamline distribution .....	60
Figure 60: Energy spectrum of the steady state solution of the Burgers equation for N=100 (DNS) .....	64

Figure 61: DNS solution for comparison .....	70
Figure 62: LES solution with $N=20$ comparison .....	71
Figure 63: Comparison of the DNS and LES solutions.....	72
Figure 64: LES solution with $N=20$ and $Ck = 0.18$ .....	72
Figure 65: Comparison of the DNS and LES solutions.....	72
Figure 66: Square cylinder problem geometry .....	73
Figure 67: Mesh representation .....	76
Figure 68: Streamline distribution .....	77
Figure 69: Pressure map .....	78
Figure 70: Velocity field .....	78
Figure 71: Streamline distribution for the 120x40 mesh.....	78
Figure 72: Streamline distribution for the 300x150 mesh.....	79
Figure 73: Pressure map for the 300x150 mesh .....	79
Figure 74: Velocity fields for the 120x40 and 300x150 mesh.....	79
Figure 75: Streamline distribution for the 120x40 mesh.....	80
Figure 76: Streamline distribution for the 300x150 mesh .....	81
Figure 77: Pressure map for the 300x150 mesh .....	81
Figure 78: Velocity fields for the 120x40 and 300x150 mesh.....	81
Figure 79: Streamline distribution for the 120x40 mesh.....	82
Figure 80: Streamline distribution for the 300x150 mesh.....	82
Figure 81: Pressure map for the 300x150 mesh .....	83
Figure 82: Velocity field for the 300x150 mesh .....	83

## Table captions

Table 1: Effect of time step on the computational time.....	25
Table 2: Solution for different number of nodes.....	26
Table 3: Other equations with a similar solution and the parameters to replace obtain it .....	29
Table 4: Value of $APe$ for low numerical schemes .....	31
Table 5: Possible values of $\Gamma/\rho$ .....	33
Table 6: Bugdet .....	85
Table 7: Schedule for the tasks .....	87



# Acknowledgements

First of all I would like to thank Professor Asensi Oliva Llena for taking me as a student under his supervision. Also he has helped me with many difficulties and problems that have raised during the last four months and provided invaluable support for my professional future. I would also like to thank Professor Xavier Trias Miquel for all the support that he has provided during this period of time, both for the seminars and for helping me with the doubts and difficulties that this project has offered. He has always found time for helping me and provided solutions to my problems and for that I am grateful. Also I would like to thank the CTTC department for all the courses and information that it has provided and also for granting me the opportunity of working in this fascinating field. Finally I would like to thank Carlos-David Pérez Segarra for being my director.

On the other hand I would like to thank my parents for their support, especially my father Daniel, for helping me with my programing skills, for providing ideas, for listening to me when I could not find the solution and for all the other things that he has done for me but cannot be written in this section for space considerations. He has provided me with invaluable support. Finally I would like to thank my colleague and friend Jordi Poblador, who provided an incredible support when I had difficulties with the driven cavity results. He has proved to be a fantastic friend.

# B1 Approach of the project

## Aim of the project

The aim of this project is to solve the differential equations that govern heat conduction and incompressible convection, mass transfer and diffusion, applied to different engineering problems using numerical methods. Therefore the following equations have to be solved:

Heat conduction:

$$\partial Q = \partial u + \partial e_c + \lambda \partial e_p + \partial W_p$$

Convection, diffusion and mass transfer: Navier-Stokes equations:

$$\frac{du}{dt} + (\mathbf{u} \cdot \nabla) \mathbf{u} = \frac{1}{Re} \Delta \mathbf{u} - \nabla p$$
$$\nabla \cdot \mathbf{u} = 0$$

When solving a problem numerically one of the most important stages is to check that the solution coincides with the real solution of the problem, and therefore that the code is correct and has no errors. Trying to experimentally check the solution of the four problems is obviously out of the scope of an end of degree project, instead a correct solution obtained with other methods shall be provided by the director for each case. Also a number of code verifications can be applied to make sure that the code is correctly implemented. These verifications consist for example in applying the laws of mass, momentum, and energy to each control volume, or in the case of turbulence checking that the symmetry of the equations is preserved. The verifications that have been implemented will be explained more precisely when they are applied.

## Scope of the project

This study has been defined by the CTTC (centre tecnològic de transferència de calor) department. The goal of this end of degree project is to obtain knowledge related to the numerical solution of heat transfer and gas dynamics applied to engineering problems. Such goal cannot be dealt with directly since it is very complex. Instead a step by step methodology will be used consisting in the solution of four simplified cases that will address the following topics: conduction, convection, linear Navier-Stokes solution and turbulence. These four topics will be studied in the following cases: Four materials conduction problem, Smith-Hutton problem, Driven cavity problem, burger's equation problem and square cylinder problem.



The scope of the project can be reduced to the following key factors:

1. Numerical solution and verification of the conduction equations
2. Numerical solution and verification of the convection equations
3. Numerical solution and verification of the Navier-Stokes equations
4. Numerical solution and verification of 1D turbulence
5. Application of the numerical solution of the Navier-Stokes equations to an aerodynamic case containing an object.

## Basic requirements

The basic requirements of a numerical simulation program can be reduced to the following factors:

- The codes developed by the student must compile and execute without any input from the executioner.
- The results must coincide with the previous simulations done by the CTTC or at least tend to the solution if the computational cost is too high to run a denser mesh.
- The code must be verifiable (for instance if a balance must be equal to 0, the code must check that the balance is in fact 0).
- The codes must be written in C++ programming language (but the results can be analysed with other programs).
- Numerical solvers cannot find an exact solution, since they implicitly have an error defined. The error must be defined small enough so that the solution is correct, in this study the numerical error is defined  $10^{-7}$ .

## Justification of the utility of the project

CFD simulations have demonstrated to be useful in a number of engineering applications. The utility of this project consists in training a student in this field so he can work in the engineering business.

### Wind tunnel simulations

Wind tunnels are probably the most exact way to determine the velocity field around an object and the forces that affect such object. The number of wind tunnels that exist in the world is very high, but many times they are for educational purposes or belong to a company and are not available for the majority of the businesses. For instance many Formula 1 teams possess wind tunnels for the design of their automobiles, but they do not share it with other companies and the prices of performing a wind tunnel experiment



can be very high. Also some wind tunnels are not automated and therefore the quality of the measurements may not be optimal. For instance one of the wind tunnels from ETSEIAT has not been automated until this semester, and is only used for educational purposes.

However many companies require a study of the flux that surrounds an object and cannot perform a wind tunnel experiment for many reasons (cost, availability...). For this reason numerical wind tunnel simulations are extremely important, and have reached a level of complexity that directly competes with wind tunnel experiments. Proof of that is that leading edge companies from the aeronautical sector use this kind of tools for the design of their products. As an example Airbus boasts of the aerodynamic advantages of designing aircrafts with CFD tools<sup>[1]</sup>.

This kind of simulations are not limited to the study of air (and other compressible fluids), but also to incompressible fluids such as water or oil. The simulations of the final case of this project resembles more the simulations that can be done in a water or oil tunnel because the equations are solved only for incompressible fluids.

These kind of simulations are especially important for hydrodynamic studies such as the study of the hydrodynamic forces on submarines or determining the drag or lift of hydrofoils.

# B2 Previous studies and state of the art

## Previous studies

All the cases that will be solved have been previously solved by the CTTC, and part of the verification process of the code consists in comparing the CTTC solution<sup>[2][3][4]</sup> with the obtained solution. This centre has realised many similar studies, both in two and three dimensions, and is one of the leading research centres in this field.

## State of the art

Gas dynamics is a subfield of fluid dynamics which is a subfield of fluid mechanics. In essence the goal of fluid mechanics is to study the physical properties and forces that affect fluids. Fluid dynamics is the study of the movement of fluids and finally gas dynamics studies the movement of gases.

The Navier-Stokes equations rule the movement of fluids. The number of applications in which the movement of fluids is very important is innumerable. The vast majority of engineering problems involve the movement of air, water or other fluids. From these applications some stand out, because the movement of fluids is the most important factor. For instance, the aeronautical sector is probably the most dependant sector on the simulation of fluids, the evolution of the aeronautical sector in the last 20 years cannot be explained without it. Still, many engineering applications ranging from fields as different as building design and construction to automobile design highly rely in the dynamic movement of air. Therefore this field has been studied very intensely, starting when the governing equations were written and continuing with the evolution of computers and numerical methods.

In fact many of the equations and methods were written long before the use of computers expanded. To take an example that will be developed in this study, the Navier Stokes equations were developed in 1822<sup>[5]</sup>, and the development of the numerical methods that are used to solve turbulent flux dates of the 1970s<sup>[6]</sup>, when computers were extremely less powerful.

As computational capacity increases the capability of solving the Navier Stokes equations numerically also increases. The applications in engineering are uncountable, there are many applications of applied aerodynamics that cannot afford a wind tunnel experiment, but still have to be designed. In this market, CFD codes are therefore very important, and if the computational cost keeps being reduced, in a few years CFD solvers will be one of the most important engineering tools.



## Navier-Stokes equations

These equations are a set of non-linear partial derivative equations, and do not have an analytical solution in the vast majority of the cases of interest. There are innumerable situations in which these equations are transcendental to understand the forces that appear on bodies that are exposed to air or other gases in movement. The most typical are applications in which solving these equations can be very useful is in the field of aerodynamics, which can be applied to problems as different as the air flux surrounding the wing of a plane, a car, or even a building.

The non-linearity of these equations implies that analytical solutions are no longer possible if the geometry is not absolutely simple. For this reason numerical solutions have been applied as a solution into solving this kind of equations.

## Numerical methods

The main concept behind numerical solution is to reduce the problem into smaller problems that can be simplified and therefore solved. There are many numerical methods, all with their advantages and disadvantages, for this kind of problems two methods have been mainly used: finite volume methods, and finite element methods. Both methods are very relevant in engineering problems, and are taught in most engineering schools. Let us analyse the main differences:

### Finite element methods

Finite element methods<sup>[7]</sup> approximate the domain into a number of sub domains or volumes. These methods represent the differential equations by functions that are applied to each control volume. The general idea is that the complex distribution function of a property can be approximated by a number of simpler functions that only activate in their element. This method implies a mathematical approach to the problem.

This kind of methods use the so called form functions, functions that are only active in the corresponding element. This method is therefore not conservative, which is a big disadvantage in this kind of problems. On the other hand they represent very well local effects, in a much more accurate way than finite volume methods, which assume that properties are constant along the surface or volume. Instead finite element methods apply a function to the element and that can be much more precise when describing local effects.

### Finite volume methods

Finite volume methods<sup>[8]</sup> approximate the domain into a number of sub domains or volumes. Each volume has a node associated to it and also some physical properties



(volume, boundary surface...). All the nodes form a mesh and this mesh then represents the whole domain. This method implies a physical approach to the problem.

In this kind of methods a property is studied in the volume, but using the Gauss theorem the integrals that are applied to each control volume can be converted to surface integrals. This can often be very useful since surface integrals imply that the flux that enters or leaves the control volume is considered and many properties are dependent on the flux. The flux that leaves a control volume through a surface is identical to the flux that enters the next control volume through that surface (for incompressible fluids), therefore this method is conservative. This is very important when solving the typical equations that appear in this kind of problems, mass, energy or momentum must be conserved and therefore a method that is conservative is very interesting.

In this case the method that is proposed to solve the equations is the finite volume method. This method is the most popular method for computational fluid dynamics, since the conservation of the properties comes pre implied. This is extremely important if the correct solution has to be reached and is the first verification a code must implement.

One of the most important tools that is used in this kind of methods is Gauss's theorem (or divergence theorem). It relates the flux of a vector field through a surface with the vector inside the volume enclosed by this surface:

$$\int_V \mathbf{v} dV = \int_S \mathbf{v} \cdot \mathbf{n} dS$$

Where  $\mathbf{v}$  is a vector field and  $\mathbf{n}$  is the normal vector of the surface.

Also it is important to be able to estimate the properties not only in the centre or the faces of the control volumes, but in any position. This is especially important because it offers the possibility of determining the value of a property in the centre knowing the value of the property at the faces or vice versa. To do this the finite difference method is used:

### Finite difference method

Finite difference methods<sup>[9]</sup> estimate the value of a property in a point knowing the value of this property in a number of nearby points. The minimum number of points that is required is two (first order scheme) and by adding more points a more precise result can be obtained. This method is based on Taylor's polynomial approximation of a function, which states that any function can be approximated as a polynomial function of order n:

$$f(x_0 + x) = f(x_0) + \frac{f'(x_0)}{1!} * x + \frac{f''(x_0)}{2!} * x^2 + \dots + \frac{f^n(x_0)}{n!} * x^n + e(x^{n+1})$$

Which can also be written as:



$$f(x_0 + x) = \sum_{0}^n \frac{f^n(x_0)}{n!} * x^n + e(x^{n+1})$$

Note that if  $n = \infty$  then the exact value of the function is obtained and that the error heavily depends on the distance between the point in which the variable is known and the point in which we want to compute it.

This theorem can also be used to obtain the value of the derivate of a function in a point when the function is known in two nearby points. For instance if an error of order 2 is neglected then the first derivative of a function is:

$$f(x_1 + d_{01}) = f(x_0) = f(x_1) + \frac{f'(x_1)}{1!} * d_{01}, \quad f(x_0) = a = \text{known}$$

$$f(x_1 + d_{12}) = f(x_2) = f(x_1) + \frac{f'(x_1)}{1!} * d_{12}, \quad f(x_2) = b = \text{known}$$

And finally, subtracting:

$$f(x_2) - f(x_0) = f(x_1) + \frac{f'(x_1)}{1!} * d_{12} - f(x_1) + \frac{f'(x_1)}{1!} * d_{01}$$

$$f'(x_1) = \frac{f(x_2) - f(x_0)}{d_{12} - d_{01}}$$

This solution is a second order central difference scheme (CDS) and higher order approximations will offer a lower error. Note that other approximations are also valid, in fact in the Smith Hutton problem some alternatives to this method are compared. In the other problems a central difference scheme shall be applied.

The current state of the art of numerical analysis of flows has reached levels of complexity that are far beyond the reach of an end of degree project. For this reason, the results that will be obtained are not necessarily new, but a way to be able to solve problems that are new.



# B3: Possible solutions and justification of the proposed solution

When trying to study a fluid dynamics problem two main possibilities arise: numerical simulation or experimentation. This study has decided to study fluid dynamics from the numerical point of view because it involves programming, offers an opportunity to enter a field with very good perspectives and which is very interesting.

The possible alternatives of numerical methods have already been discussed in the state of the art section, and the differences between finite volume methods and finite element methods are now clear. For this study the finite volume method has been chosen for the following reasons:

- The conservation of mass, energy or momentum can be easily validated at each control volume
- The implementation and the discretisation of the equations is easier
- The CTTC proposes this method

For the following 5 cases the problems will differ slightly in their equations, but not in how they have been solved: the numerical method shall always be a finite volume method, the equations have been discretised and a program is defined.

# B4 Development of the solution and justification of the results

## Conduction problem

### Introduction

The main goal of this exercise is to solve the following bi-dimensional conduction problem:

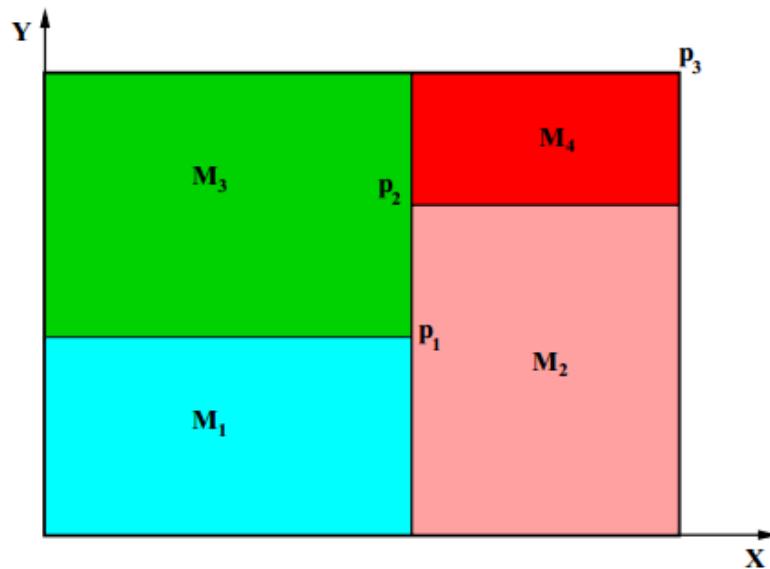


Figure 1: General schema of the proposed problem

	$x[m]$	$y[m]$
$p_1$	0.50	0.40
$p_2$	0.50	0.70
$p_3$	1.10	0.80

Table 1: Problem coordinates

	$\rho[kg/m^3]$	$c_p[J/kgK]$	$k[W/mK]$
$M_1$	1500.00	750.00	170.00
$M_2$	1600.00	770.00	140.00
$M_3$	1900.00	810.00	200.00
$M_4$	2500.00	930.00	140.00

Table 2: Physical properties

Cavity wall	Boundary condition
Bottom	Isotherm at $T = 23.00^\circ\text{C}$
Top	Uniform $Q_{flow} = 60.00 \text{W/m length}$
Left	In contact with a fluid at $T_g = 33.00^\circ\text{C}$ and heat transfer coefficient $9.00 \text{W/m}^2\text{K}$
Right	Uniform temperature $T = 8.00 + 0.005t^\circ\text{C}$ (where $t$ is the time in seconds)

Table 3: Boundary conditions

Figure 1: Physical problem

The tool used to solve this problem has been a C++ program named CRESPO.dat, which generates two data files: the first is designed to be plotted with the gnuplot tool to obtain a representation of the temperature field, the latter is the file that contains the solution of the problem and includes the temperature at the given coordinates.

## Application of the conduction laws:

The problem that has to be solved is the conduction problem. The conduction law can be written as:

$$\partial Q = \partial u + \partial e_c + \lambda d e_p + \partial W_p$$

In a purely conductive problem:

$$\partial Q = \partial u$$

$$Q = -\lambda \nabla T$$

Therefore in each direction:

$$q_x = -\lambda \frac{\partial T}{\partial x}, \quad q_y = -\lambda \frac{\partial T}{\partial y}, \quad q_z = -\lambda \frac{\partial T}{\partial z}$$

In a bi-dimensional problem  $q_z$  is not considered and therefore for a control volume the following expressions can be written:

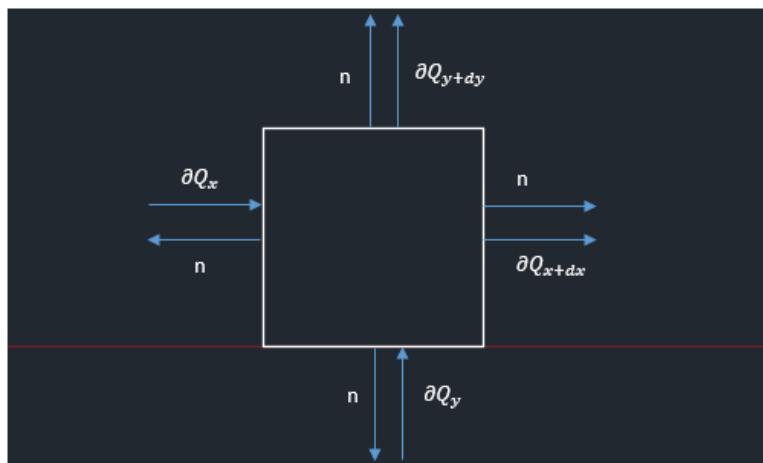


Figure 2: differential control volume



And therefore:

$$\frac{\partial Q_x}{\partial t} = -\lambda \frac{\partial T}{\partial x} \partial y$$

$$\frac{\partial Q_{x+dx}}{\partial t} = \frac{\partial}{\partial x} \left( -\lambda T - \lambda \frac{\partial T}{\partial x} \partial x \right) \partial y = \left( -\lambda \frac{\partial T}{\partial x} - \lambda \frac{\partial^2 T}{\partial x^2} \partial x \right) \partial y$$

$$\frac{\partial Q_y}{\partial t} = -\lambda \frac{\partial T}{\partial y} \partial x$$

$$\frac{\partial Q_{y+dy}}{\partial t} = \frac{\partial}{\partial y} \left( -\lambda T - \lambda \frac{\partial T}{\partial y} \partial y \right) \partial x = \left( -\lambda \frac{\partial T}{\partial y} - \lambda \frac{\partial^2 T}{\partial y^2} \partial y \right) \partial x$$

If a balance of Q is applied in each direction:

$$\frac{\partial Q_x}{\partial t} - \frac{\partial Q_{x+dx}}{\partial t} = -\lambda \frac{\partial T}{\partial x} \partial y - \left( -\lambda \frac{\partial T}{\partial x} - \lambda \frac{\partial^2 T}{\partial x^2} \right) \partial y = \lambda \frac{\partial^2 T}{\partial y^2} \partial x \partial y$$

$$\frac{\partial Q_y}{\partial t} - \frac{\partial Q_{y+dy}}{\partial t} = -\lambda \frac{\partial T}{\partial y} \partial x - \left( -\lambda \frac{\partial T}{\partial y} - \lambda \frac{\partial^2 T}{\partial y^2} \right) \partial y = \lambda \frac{\partial^2 T}{\partial x^2} \partial x \partial y$$

$$\partial Q = \lambda \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \partial x \partial y \partial t$$

The main assumption that has not been commented yet is that  $\lambda$  must be constant in the differential.

Assuming that the physical properties are constant:

$$\partial u = \frac{\partial u}{\partial t} \partial t = \rho \partial v C_v \frac{\partial T}{\partial t}$$

$$\lambda \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \partial x \partial y \partial t = \rho \partial v C_v \frac{\partial T}{\partial t}$$

$$\frac{\partial T}{\partial t} = \frac{\lambda}{\rho C_v} \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)$$

To demonstrate the formula of the conductivity between two materials of different conductivities is unnecessary. As can be demonstrated it follows a harmonic mean:

$$\lambda_{12} = \frac{2}{\frac{1}{\lambda_1} + \frac{1}{\lambda_2}}$$

## Problem solution:

To solve the problem numerically the approach that has been used is to establish first a basic division of the field in order to simplify the problem.  $\lambda$  will be constant in all the elements, with this goal in mind a basic square element of size  $0,1\text{m} \times 0,1\text{m}$  has been used. The  $0,1\text{m}$  size in the x axis has been chosen because in the x coordinate the field is  $1,1\text{m}$  long and the material division is situated at  $0,5\text{m}$  and therefore the element position could fit correctly in the given space. On the other hand the  $0,1\text{m}$  has been chosen for a similar reason, the total length of the field is  $0,8\text{m}$  and the divisions are situated at  $0,4\text{m}$  and  $0,7\text{m}$ , as shown in Figure 3.

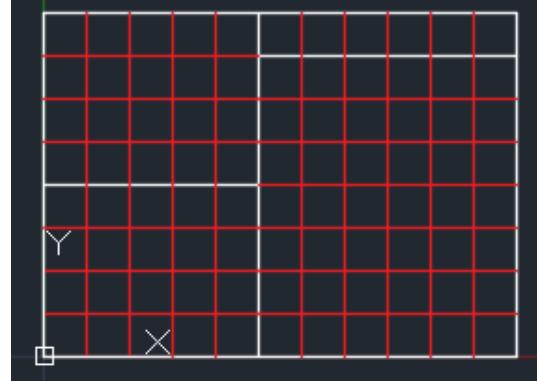


Figure 3: First element division

Inside these elements a second division is applied, each element is divided in  $n_x$  elements in the x axis and  $n_y$  elements in the y axis. In this case the chosen number of elements is:  $n_x=10$ ,  $n_y=10$ . This second division can be seen in Figure 4.

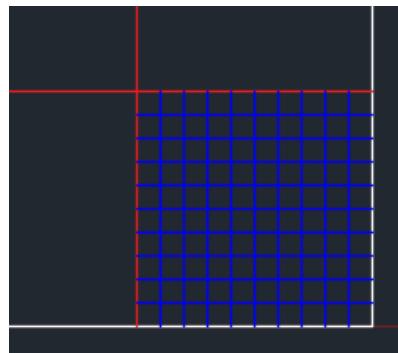


Figure 4: Second element division

## Generic elements:

For a generic element the geometry shall be:

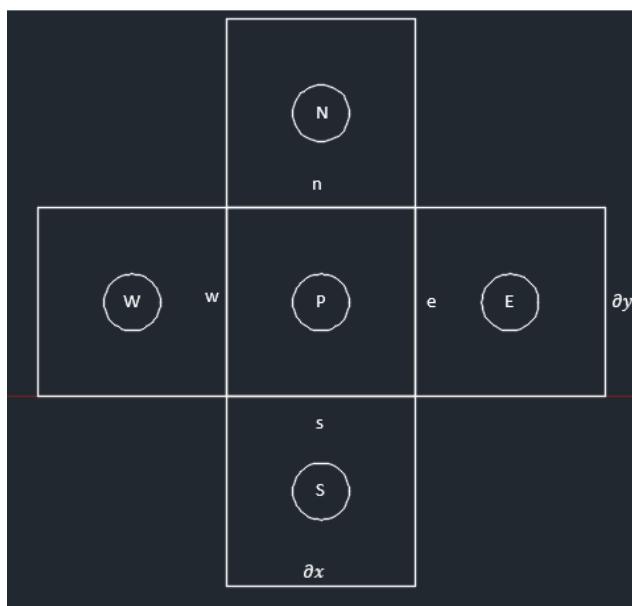


Figure 5: generic node description



The discretised form shall be:

$$\dot{Q}_P = \dot{Q}_w - \dot{Q}_e + \dot{Q}_s - \dot{Q}_n = -\lambda \left( \frac{\partial T}{\partial x} \right)_w + \lambda \left( \frac{\partial T}{\partial x} \right)_e - \lambda \left( \frac{\partial T}{\partial y} \right)_s - \lambda \left( \frac{\partial T}{\partial y} \right)_n$$

$$\dot{Q}_P = \int_{t^n}^{t^{n+1}} \rho C v \left( \frac{\partial T}{\partial t} \right)_P$$

Since the first order derivate can be written as:

$$\left( \frac{\partial T}{\partial x} \right)_e = \frac{T_E - T_P}{d_{E-P}} = \frac{T_E - T_P}{\Delta x_E}, \quad \left( \frac{\partial T}{\partial x} \right)_w = \frac{T_P - T_W}{d_{W-P}} = \frac{T_P - T_W}{\Delta x_W}$$

$$\left( \frac{\partial T}{\partial x} \right)_s = \frac{T_S - T_P}{d_{S-P}} = \frac{T_S - T_P}{\Delta y_S}, \quad \left( \frac{\partial T}{\partial x} \right)_n = \frac{T_P - T_N}{d_{N-P}} = \frac{T_P - T_N}{\Delta y_N}$$

$$\int_{t^n}^{t^{n+1}} \left( \frac{\partial T}{\partial t} \right)_P = \frac{T^{n+1} - T^n}{\Delta t}$$

Finally adding a relaxation factor:

$$\begin{aligned} \rho C v \frac{T^{n+1} - T^n}{\Delta t} &= \beta \dot{Q}_P^{n+1} - (1 - \beta) \dot{Q}_P^n \\ \rho C v V \frac{T^{n+1} - T^n}{\Delta t} &= \lambda \beta \left( \frac{T_E - T_P}{\Delta x_E} S_e - \frac{T_P - T_W}{\Delta x_W} S_w + \frac{T_S - T_P}{\Delta y_S} S_s - \frac{T_P - T_N}{\Delta y_N} S_n \right)^{n+1} \\ &\quad - \lambda (1 - \beta) \left( \frac{T_E - T_P}{\Delta x_E} S_e - \frac{T_P - T_W}{\Delta x_W} S_w + \frac{T_S - T_P}{\Delta y_S} S_s - \frac{T_P - T_N}{\Delta y_N} S_n \right)^n \end{aligned}$$

Finally a more compact expression is:

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

Where:

$$a_P = \rho V \frac{Cv}{\Delta t} + \lambda \beta \left( \frac{S_e}{\Delta x_E} + \frac{S_w}{\Delta x_W} + \frac{S_s}{\Delta y_S} + \frac{S_n}{\Delta y_N} \right)$$

$$a_E = \beta \frac{S_e}{\Delta x_E}, a_W = \beta \frac{S_w}{\Delta x_W}, a_S = \beta \frac{S_s}{\Delta y_S}, a_N = \beta \frac{S_n}{\Delta y_N}$$

$$\begin{aligned} b_P^{n+1} &= (a_E (T_E^n - T_P^n) + (a_W T_W^n - T_P^n) + (a_N T_N^n - T_P^n) + (a_S T_S^n - T_P^n))(1 - \beta) \\ &\quad + \rho V \frac{Cv}{\Delta t} T_P^n \end{aligned}$$

Or in a more compact notation:

$$b_P^{n+1} = (1 - \beta)\dot{Q}_P^n + \rho V \frac{Cv}{\Delta t} T_P^n$$

### Boundary nodes

For a more precise solution an additional node has been added on the boundary of the problem, its geometry changes, and should be taken into account:

Boundary nodes do not have a control volume of their own and the distance between a boundary node and the generic node that is closest to them is  $\Delta x/2$  or  $\Delta y/2$ .

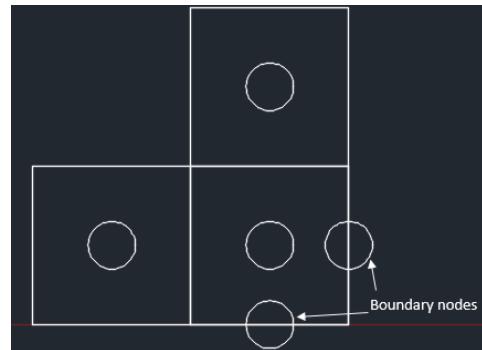


Figure 6: Boundary node description

### Determination of the coefficients

Now to solve the problem the only difficulty is to determine the different coefficients.

In 2D the surfaces for a generic node can be calculated as:

$$S_e = S_w = \Delta y = 0,1/n_y$$

$$S_n = S_s = \Delta x = 0,1/n_x$$

$$V = \Delta x \Delta y = \frac{0,1^2}{n_x n_y}$$

$\lambda, \rho, Cv, T^0$  are all data, and depend on the material.

To choose the time step a characteristic time can be calculated from the thermal conductivity of the different materials (The dimensionless number for conduction is the Fourier number):

$$Fo = \frac{\alpha t}{L^2}$$

Typical values of the Fourier number can vary between 0.5 and 2 (see bibliography). To get an idea of the order of magnitude of the characteristic time the value  $Fo=1$  has been supposed in the program.

$$t = Fo \frac{L^2}{\alpha}, \quad \alpha = \frac{\lambda}{\rho C_p}$$

This characteristic time can be computed for each material. To correctly apply this time, the time step must be equal or lower than the lowest characteristic time of the 4 materials.



$$\Delta t \leq t$$

### Boundary conditions applied to the coefficients:

The program must compute each coefficient for each node and store them. For a generic node the functions calcap computes  $a_P$ , calcax computes  $a_N, a_S, a_W, a_E$  (taking advantage that the formulas are very similar) and calcbp computes  $b_p$ .

To impose the boundary conditions the function calculscoef is implemented, for each boundary condition it establishes how the coefficients shall differ from the generic node coefficients. The function condfrontera will alter the value of the temperature of those nodes that have the temperature of their nodes determined by a boundary condition.

#### Material 1:

The left wall ( $x=0$ ) is in contact with a fluid of known  $\alpha$  and temperature. Therefore the left wall nodes shall have an additional term in the bp term that accounts for the convective term:

$$b_P = b_P + \alpha(T_g - T_P^n)$$

The bottom wall ( $y=0$ ) is isothermal and therefore:  $T = T_{bot}$

The right ( $x=0,5$ ) and top ( $y=0,4$ ) walls are in contact with materials 2 and 3 respectively and shall have their conductivity terms altered:

$$\lambda_r = \lambda_{12}, \quad \lambda_t = \lambda_{13}$$

#### Material 2:

The right wall ( $x=1,1$ ) has a uniform temperature that varies with time. Therefore:

$$T = T_{right}$$

The bottom wall ( $y=0$ ) is isothermal and therefore:

$$T = T_{bot}$$

The left ( $x=0,5$ ) is in contact with materials 1 ( $y < 0,4$ ) and 3 ( $0,4 < y < 0,7$ ) and the top ( $y=0,7$ ) wall is in contact with material 3. They shall have their conductivity terms altered:

$$\lambda_l = \lambda_{12} \text{ } (y < 0,4), \quad \lambda_l = \lambda_{23} \text{ } (0,4 < y < 0,7), \quad \lambda_t = \lambda_{13}$$

#### Material 3:

The left wall ( $x=0$ ) is in contact with a fluid of known  $\alpha$  and temperature. Therefore the left wall nodes shall have an additional term in the bp term that accounts for the convective term:

$$b_P = b_P + \alpha(T_g - T_P^n)$$



The top wall ( $y=0,8$ ) has a uniform heat flux and therefore its nodes shall have an additional term in the  $b_P$  term that accounts for the convective term:

$$b_P = b_P + Q_{top}$$

The right ( $x=0,5$ ) is in contact with materials 2 ( $0,4 < y < 0,7$ ) and 3 ( $0,7 < y < 0,8$ ) and the bottom ( $y=0,4$ ) wall is in contact with material 3. They shall have their conductivity terms altered:

$$\lambda_r = \lambda_{23} \quad (0,4 < y < 0,7), \quad \lambda_r = \lambda_{34} \quad (0,7 < y < 0,8), \quad \lambda_b = \lambda_{13}$$

#### Material 4:

The right wall ( $x=1,1$ ) has a uniform temperature that varies with time. Therefore:

$$T = T_{right}$$

The top wall ( $y=0,8$ ) has a uniform heat flux and therefore its nodes shall have an additional term in the  $b_P$  term that accounts for the convective term:

$$b_P = b_P + Q_{top}$$

The left ( $x=0,5$ ) and bottom ( $y=0,4$ ) walls are in contact with materials 3 and 2 respectively and shall have their conductivity terms altered:

$$\lambda_l = \lambda_{24}, \quad \lambda_b = \lambda_{34}$$

For all generic nodes that are next to a boundary node the distance from such node is half the distance it would have from another node in the same direction. The same applies for boundary node and the distance to the closest generic node. This shall alter  $a_N, a_S, a_W$  and/or  $a_E$  depending on which direction the distance is reduced.

#### Main program

The main program establishes the basic parameters such as the time step and the total time to simulate. It initiates the matrixes where the initial temperature is stored and the coordinates of the nodes. The coefficients are constant throughout the iterative process and therefore can be computed only once at the start of the program.

Then the iterative process starts and the program computes a new temperature for each node with the given equations for time=0s:

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

This temperature is compared with the previous temperature in each node and while the difference between these two temperatures on any node is higher than the given permissible error the iterative process is repeated:



$$\max(\text{abs}(T_P^{n+1} - T_P^{n+1*})) \leq \text{error}$$

When such difference is lower than the error the program determines if the next time step has to be computed and repeats the process with the computed temperature as the old temperature:

$$T_P^n = T_P^{n+1}$$

And so on until no more time steps have to be computed.

The main parameters of the program can be altered at the parameter definition, for instance in this case a Crank-Nicolson scheme has been used to ensure that the solution did converge without reducing the time steps to an unnecessary level.

### General program scheme

This program can use both an implicit, explicit or intermediate schemes to solve the conduction equations. The solution of the system of equations:

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

Note that some coefficients are simplified for the explicit and implicit methods:

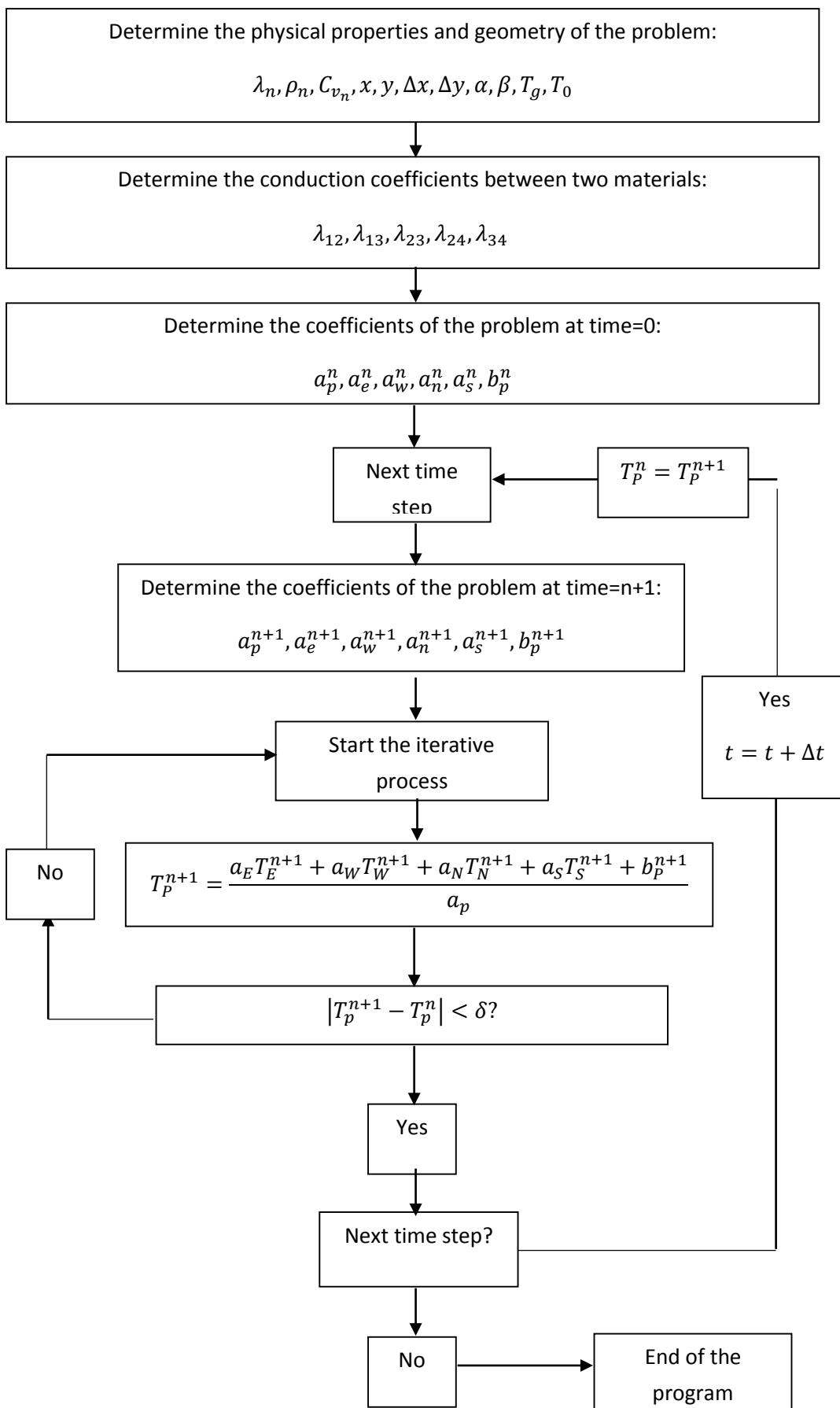
For implicit methods ( $\beta = 0$ ) the coefficients  $a_e, a_w, a_n, a_s$  are all equal to 0. This also affects  $a_p$ , that shall be simplified to  $\rho V \frac{Cv}{\Delta t}$ . These simplifications also affect the system of equations, since in implicit methods the solution at time n+1 only depends on the values at the previous time step (n). Therefore the equation becomes:

$$a_P T_P^{n+1} = b_P^{n+1}$$

Note that  $b_P^{n+1}$  only depends on the previous time step conditions. This makes the solution of the problem much faster since an iterative process is not necessary. On the other hand this process has convergence problems and may require a smaller time step.

For explicit methods ( $\beta = 1$ ) the solution at the current time step (n+1) depends only on the current time step conditions (n+1). In this case the simplifications are much less significant, only the term  $b_p$  is simplified to  $b_P^{n+1} = \rho V \frac{Cv}{\Delta t} T_P^n$ .

This system of equations can be solved in many ways, in this case a gauss-seidel method has been used. Therefore the general scheme of the problem shall be:



### Program output functions:

The output of the functions is determined by two functions: `treuredades` shall open a data file that prints the temperature of every node as well as its coordinates in a fashion that the program `gnuplot` can use to plot the temperature distribution graphically. An example of these plots can be:

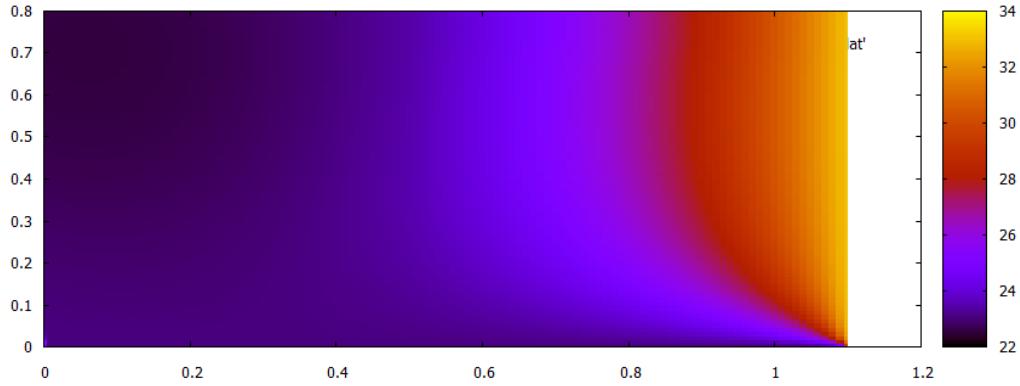


Figure 7: Temperature distribution at time=5000s

On the other hand `calcentrega` determines the temperature at the requested coordinates. Since these coordinates may not correspond with a node the function determines the closest nodes to the coordinates and then interpolates the temperature between such nodes.

To interpolate the temperature the minimum order function that is required is:

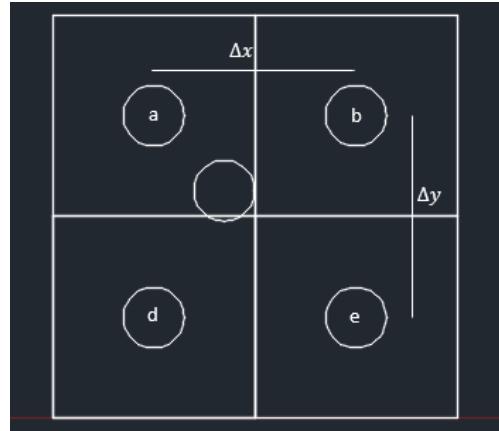


Figure 8 Node interpolation scheme

$$T = a_0x + a_1y + a_2xy + a_3$$

And therefore the system is a 4 equation, 4 unknowns:

$$T_a = a_0x_W + a_1y_N + a_2x_Wy_N + a_3$$

$$T_b = a_0x_E + a_1y_N + a_2x_Ey_N + a_3$$

$$T_c = a_0x_E + a_1y_S + a_2x_Ey_S + a_3$$

$$T_d = a_0x_W + a_1y_S + a_2x_Wy_S + a_3$$

Using maple to solve the system:

$$a_0 = (y_S * T_a - y_S * T_b + y_N * T_c - y_N * T_d) / (y_N - y_S) / (x_E - x_W);$$

$$a_1 = (x_E * T_a - x_W * T_b + x_W * T_c - x_E * T_d) / (y_N - y_S) / (x_E - x_W);$$

$$a_2 = -(T_a - T_b + T_c - T_d) / (x_E * y_N - x_E * y_S - x_W * y_N + x_W * y_S);$$

$$a_3 = -(T_a * x_E * y_S - T_b * x_W * y_S + T_c * x_W * y_N - T_d * x_E * y_N) / (y_N - y_S) / (x_E - x_W);$$

The output of the function is the temperature at the given coordinates.

## Results:

### Temperature temporal evolution

The solution of this problem is a transient solution since the right wall's temperature is a function of time. Therefore the temperature distribution has two main phases:

During the first phase heat enters mainly through the bottom wall and is mainly dissipated through the left wall. At these early stages the amount of heat that is dissipated is very small since the fluid temperature is very similar to the left wall temperature. The bottom wall is fixed at  $T_{bot}$  which is higher than the mean temperature of the materials. Therefore it can be seen that the temperature of the inferior region of the studied area rises:

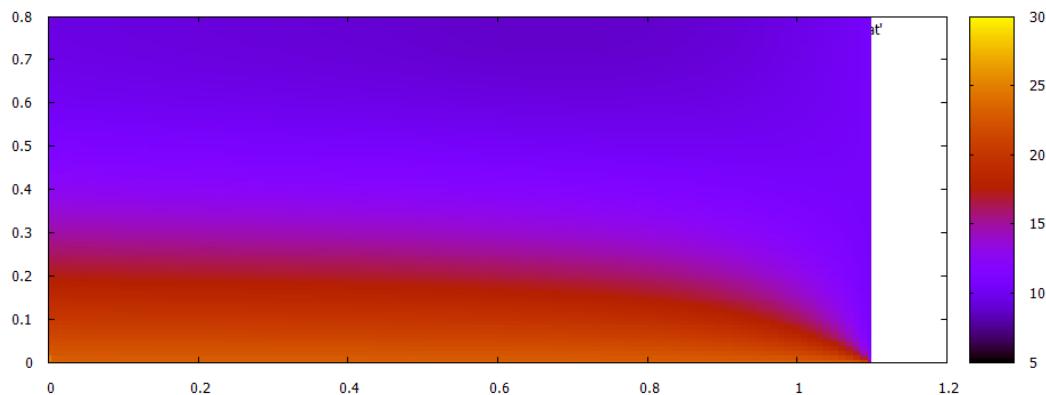


Figure 9: Temperature distribution at time=500s (first phase)

This behaviour is maintained until  $t \approx 2500s$ , then the right wall's temperature begins to rise into a temperature that is of the same order as the bottom wall's temperature. This leads to the second phase (which shall be explained later) after a transient state in which both the bottom and right walls are introducing heat and only the left wall is dissipating heat.

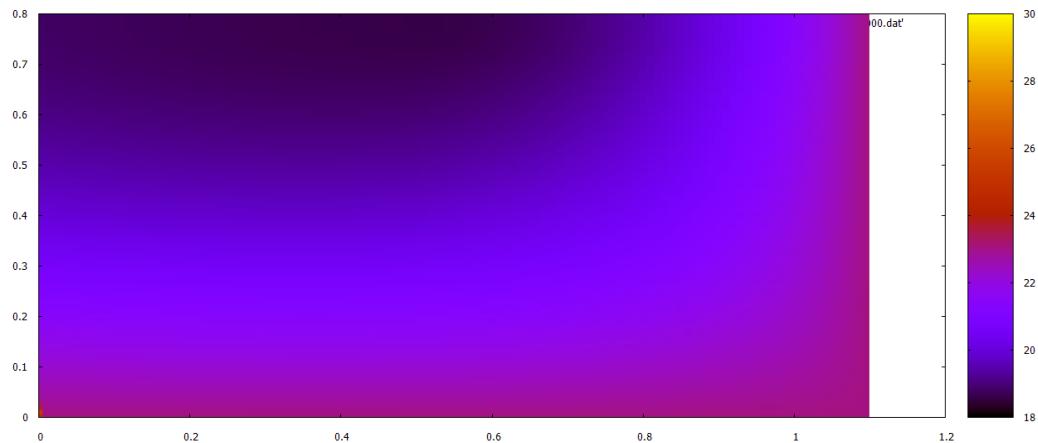


Figure 10: Temperature distribution at time=3000s (transient state)

During the second phase heat enters mainly through the right wall and is dissipated through the left and bottom walls. The temperature of the right wall rises with time and reaches values that are high enough so that the bottom wall has to dissipate heat in order to maintain its temperature. As in both previous phases the top wall does not introduce a relevant amount of heat, but it does not withdraw a relevant amount of heat either. Therefore it acts similarly to an adiabatic wall. Then a sail type pattern appears. With increasing time the width and the temperature of the sail pattern increase due to the increasing temperature of the right wall. This can be seen in the following figures.

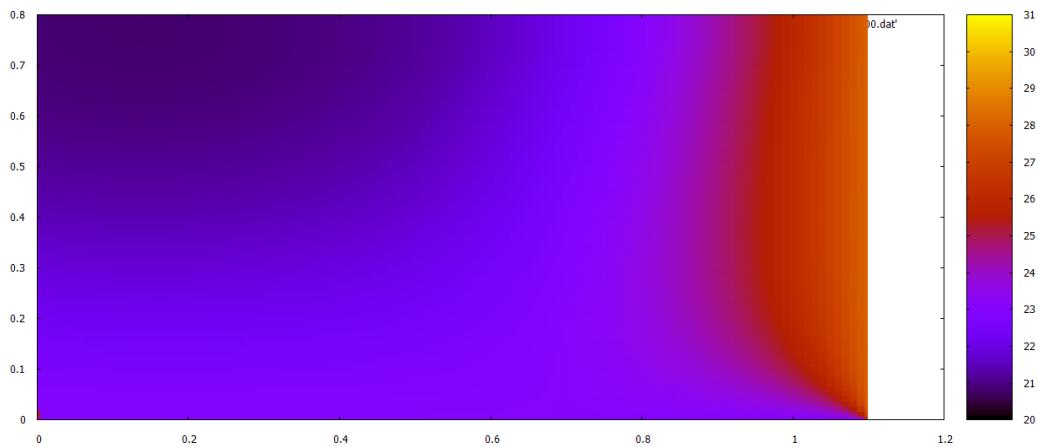


Figure 11: Temperature distribution at time=4000s (second phase)

The results for this problem are given at time=10000s, so to compare the solution accurately 10000s have to be computed. Then the temperature distribution at the last simulated instant will be:

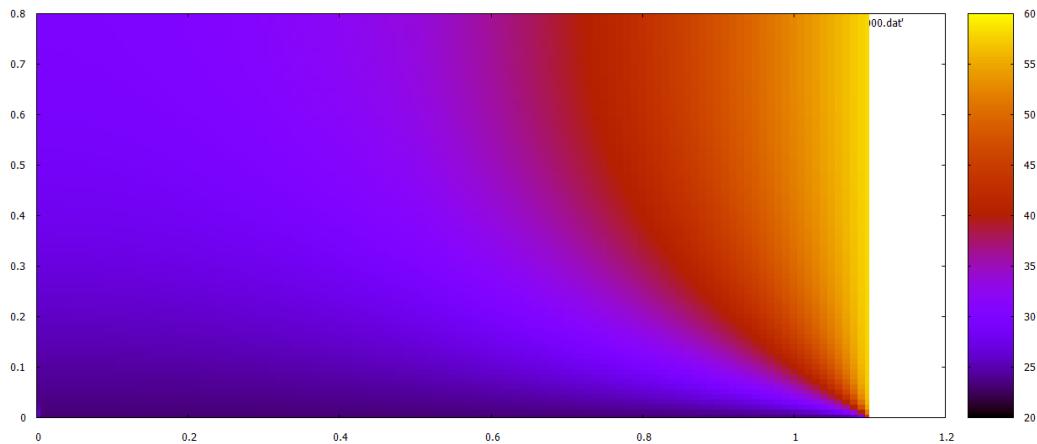


Figure 12: Temperature Distribution at time=10000s

Note that all these solutions have been computed for a number of subdivisions of 10 for each block both in the x and y axis. Taking into account that there are 11 blocks in the horizontal axis and 8 blocks in the vertical axis, the computational cost of this solution has been of 112x82 (9184 nodes in total). The computational time step is 10s.

The whole temporal evolution of the problem can be perceived in the animation that has been designed with gnuplot. This animation is available online in the following link: <https://www.youtube.com/watch?v=bzUiKy7lcdM&feature=youtu.be>.

### Contour plot

It is also interesting to plot the contour plots of the temperature. This kind of plots offer the possibility to visualise all the points that share the same temperature joint by a line. In Figure 13 the temperature differences between each line are of 1°C. It is clear then that the temperature gradient is much more important near the right wall, especially at the bottom of that wall.

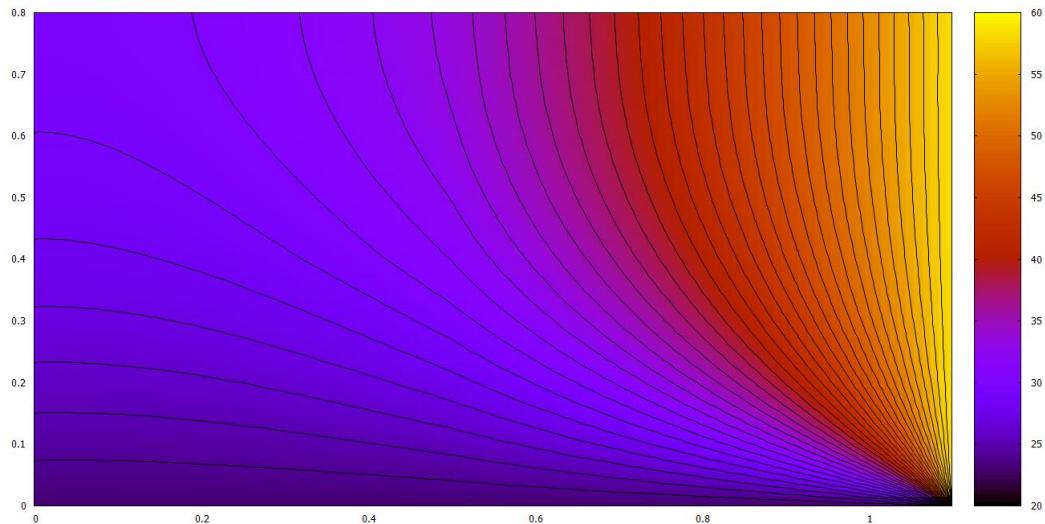


Figure 13: Contour plot of the solution at time=10000s

### Effect of time step

Another important aspect in transient problems in particular and in numerical schemes in general is the time step. Small time steps improve the convergence of the solution, but also increase the computational cost. Larger time steps will reduce the computational cost, but if the chosen time step is too large then the solution may not converge.

Still, it is interesting to compare the computational time of the solution taking into account different time steps. More accurate solutions (lower time steps) shall require a higher computational cost and therefore the time will increase dramatically. Note that the following values have all been obtained with a mesh density of 10 nodes per block (112x82nodes).

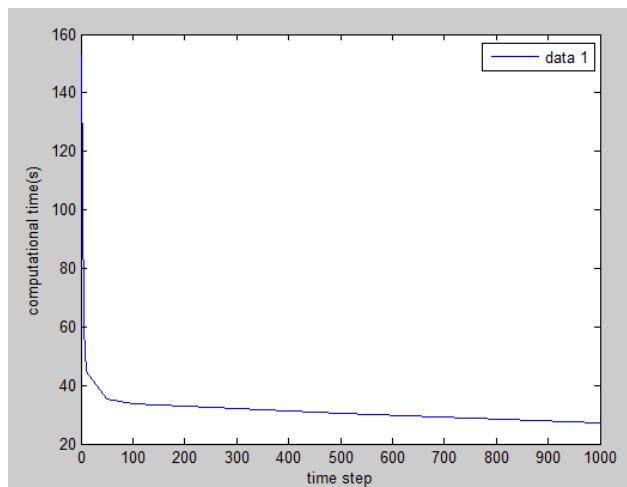


Figure 14: Effect of the time step in the computational time

This data cannot even be resembled to a logarithmic plot, this proves the strong tendency of the solution on the time step. It seems very clear that using a time step lower than 5 is unviable.

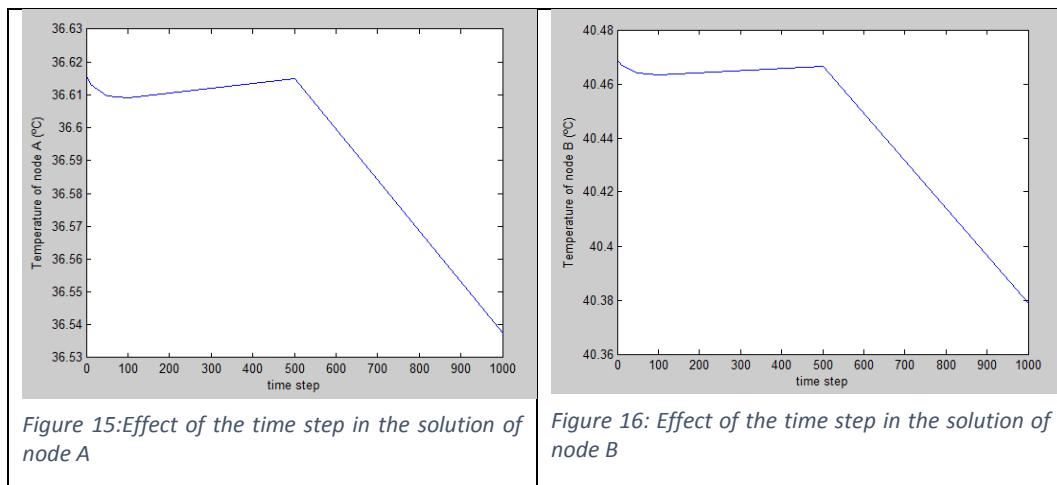


Figure 15: Effect of the time step in the solution of node A

Figure 16: Effect of the time step in the solution of node B

By looking at the temperature variation with the time step it seems clear that the time step is not very relevant when studying the accuracy of the solution. Even with very high time step values ( $>500$ ) the solution shall only vary in a percentage of about 0.5% ( $0.1^\circ\text{C}$ ). This tendency is present in the solution of both node A (Figure 15) and node B (Figure 16).

Time (s)	$T_A$	$T_B$	Computational time	Time step (s)
10000	36,5376	40,3789	27,34	1000
10000	36,6149	40,4664	30,3	500
10000	36,6091	40,4633	33,84	100
10000	36,6098	40,4639	35,34	50
10000	36,6133	40,4667	44,84	10
10000	36,6146	40,4678	59,3	5
10000	36,6154	40,4686	153,3	1

Table 1: Effect of time step on the computational time

Note that to plot these results a Matlab code has been implemented, it can be found at the annex section.

### Effect of mesh density

The same result than in the case of reducing the time step is expected for increasing the density of the mesh: the accuracy of the solution shall increase at expense of the computational time. In the following figures the effect of the variation of the mesh density is studied. Note that the when studying the computational time this value does not have a meaning, since for different computers this value shall be completely different. The interest of computing it is to establish the tendency of the time evolution, not the value itself. Also note that the following values have all been obtained with a time step of 10s.

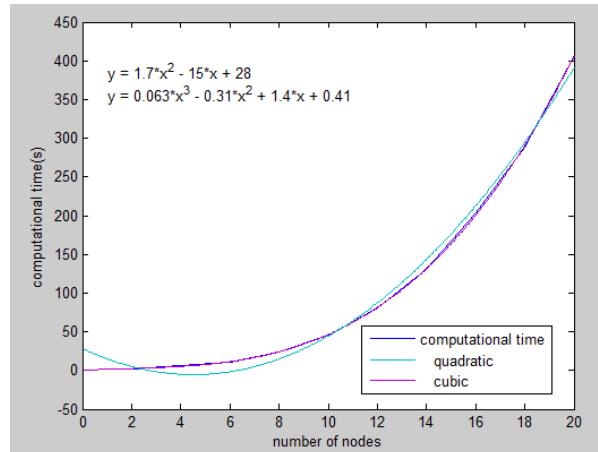


Figure 17: Computational time trend lines

As expected, by increasing the number of nodes the computational cost increases as  $n^2$  (bidimensional problem) and the computational time increases at an even higher rate which is closer to a cubic tendency (from Figure 17 it is clear that the cubic trend line (in pink) is much more accurate than the quadratic trend line (in light blue)). This behaviour is normal in this kind of solutions since there are many factors that contribute to increase the computational time. Some examples of these factors could be compiler errors, memory usage, etc.

Now the accuracy of the solution shall be analysed:

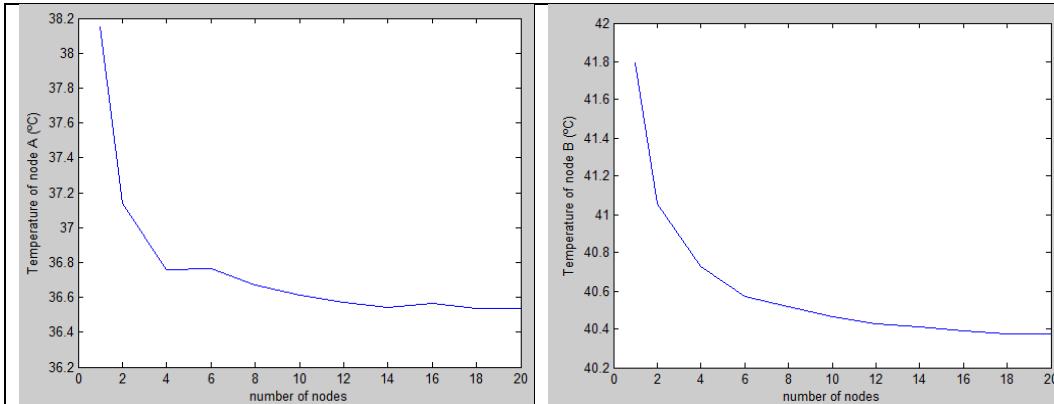


Figure 18: Effect of the number of nodes on the solution of node A

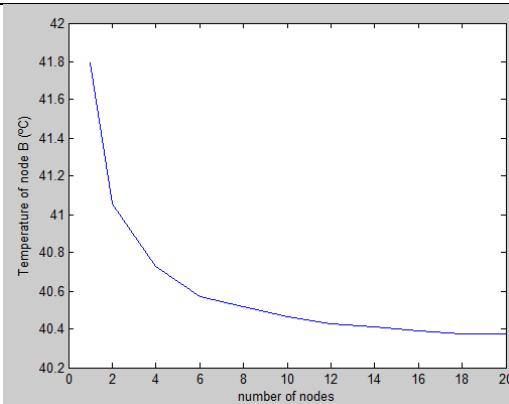


Figure 19: Effect of the number of nodes on the solution of node B

Also as expected, the accuracy of the solution increases dramatically with the number of nodes. In Figure 18 it seems very clear that the solution tends to reach a temperature value of approximately 36.54°C (see Table 2). The same happens in Figure 19, the solution clearly tends to approximately 40.38°C (see Table 2).

Time (s)	$T_A$ (°C)	$T_B$ (°C)	Computational time (s)	Number of nodes per block
10000	38,15	41,79	1,59	1
10000	37,14	41,06	2,15	2
10000	36,76	40,73	5,80	4
10000	36,77	40,57	11,18	6
10000	36,67	40,52	24,45	8
10000	36,61	40,47	45,55	10
10000	36,57	40,43	81,21	12
10000	36,54	40,42	129,90	14
10000	36,57	40,39	204,30	16
10000	36,54	40,38	288,20	18
10000	36,54	40,38	406,60	20

Table 2: Solution for different number of nodes

Note that to plot, compare and analyse these results a Matlab code has been implemented, it can be found at the corresponding annex section.

# The Smith Hutton problem

## Introduction

The main goal of this exercise is to solve the following bi-dimensional convection-diffusion problem:

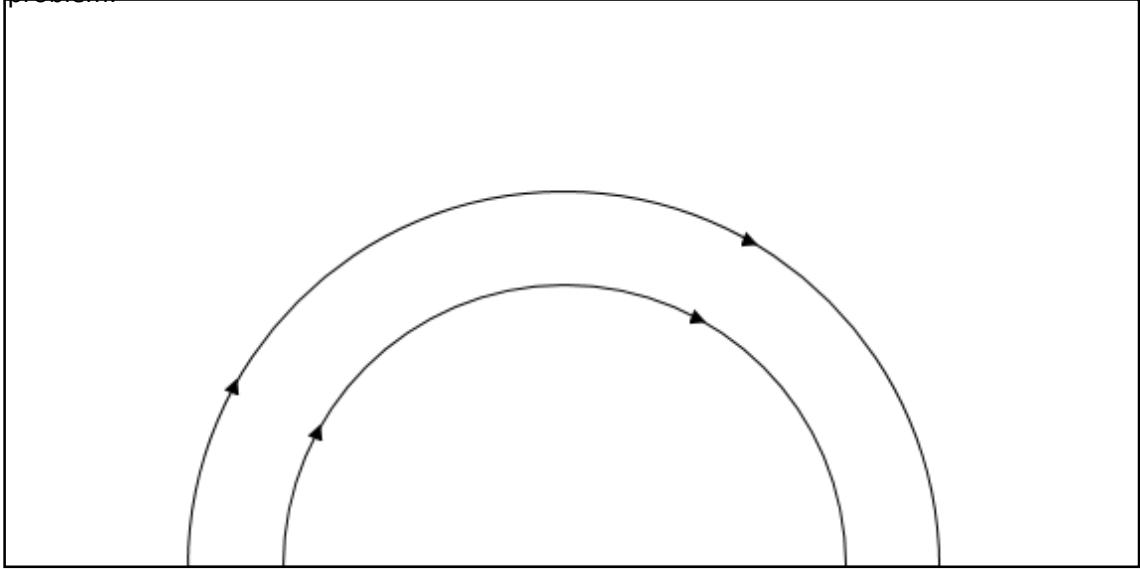


Figure 20: Smith Hutton problem definition

The two-dimensional convection-diffusion equation is:

$$\frac{d(\rho\phi)}{dt} + \nabla(\rho v\phi) = \nabla(\Gamma\nabla\phi) + S$$

In the rectangular domain the prescribed velocity field is:

$$u(x, y) = 2y(1 - x^2)$$

$$v(x, y) = 2x(1 - y^2)$$

And the boundary conditions for the variable  $\phi$  are:

$$\phi = 1 + \tanh(\alpha(2x + 1)), \quad y = 0, x \in (-1, 0) \text{ (inlet)}$$

$$\frac{d\phi}{dy} = 0, \quad y = 0, x \in (0, 1) \text{ (outlet)}$$

$$\phi = 1 - \tanh(\alpha) \text{ (elsewhere)}$$

And  $\alpha = 10$ .

The dimensions of the problem are 2x1.



In this problem the velocity field is prescribed and the goal is to determine the variable  $\phi$  distribution. The goal is to design a CFD code that correctly solves the convection diffusion equation.

## Application of the convection diffusion laws:

The two-dimensional convection-diffusion equation is:

$$\frac{d(\rho\phi)}{dt} + \nabla(\rho v\phi) = \nabla(\Gamma\nabla\phi) + S$$

Integrating the differential convection-diffusion equation, then applying the gauss theorem and finally discretising the equation for a rectangular mesh the discretised equation can then be written as:

$$\begin{aligned} & \frac{(\rho\phi)_P^{n+1} - (\rho\phi)_P^n}{\Delta t} \Delta x \Delta y + [(\rho\phi)_e^{n+1} - (\rho\phi)_w^{n+1}] \Delta y + [(\rho\phi)_n^{n+1} - (\rho\phi)_s^{n+1}] \Delta x \\ &= \left[ \left( \Gamma \frac{d\phi}{dx} \right)_e^{n+1} - \left( \Gamma \frac{d\phi}{dx} \right)_w^{n+1} \right] \Delta y + \left[ \left( \Gamma \frac{d\phi}{dx} \right)_n^{n+1} - \left( \Gamma \frac{d\phi}{dx} \right)_s^{n+1} \right] \Delta x \\ &+ S_P^{n+1} \Delta x \Delta y \end{aligned}$$

In this equation it is interesting for the results analysis to identify the following terms:

Temporal term: Introduces the temporal evolution of property  $\phi$

$$\frac{(\rho\phi)_P^{n+1} - (\rho\phi)_P^n}{\Delta t} \Delta x \Delta y$$

Convective term: Introduces the concept of convection which is the variation of the property  $\phi$  due to the movement of the fluid particles:

$$[(\rho\phi)_e^{n+1} - (\rho\phi)_w^{n+1}] \Delta y + [(\rho\phi)_n^{n+1} - (\rho\phi)_s^{n+1}] \Delta x$$

Diffusive term: Introduces the concept of convection which is the variation of the property  $\phi$  due to the difference in concentration of the property  $\phi$  in the different regions of the domain:

$$\left[ \left( \Gamma \frac{d\phi}{dx} \right)_e^{n+1} - \left( \Gamma \frac{d\phi}{dx} \right)_w^{n+1} \right] \Delta y + \left[ \left( \Gamma \frac{d\phi}{dx} \right)_n^{n+1} - \left( \Gamma \frac{d\phi}{dx} \right)_s^{n+1} \right] \Delta x$$

Source term: Introduces any sources of  $\phi$  that there may be in the domain. In this case the source term shall be 0 in all the domain:

$$S_P^{n+1} \Delta x \Delta y = 0$$

It is interesting to note that these equations can be applied to various governing equations, by only substituting the terms  $\phi, \Gamma$  and  $S$  by their homologs:

Equation	$\phi$	$\Gamma$	$S$
<b>Continuity</b>	1	0	0
<b>Momentum in x direction</b>	u	$\mu$	$-dp_d/dx$
<b>Momentum in x direction</b>	v	$\mu$	$-dp_d/dx \rho g \beta (T - T_\infty)$
<b>Energy (constant Cp)</b>	T	$\lambda/C_p$	$\phi/C_p$

Table 3: Other equations with a similar solution and the parameters to replace obtain it

Therefore with a very similar solver all these equations can be solved.

Finally, prior to the solution of the convection-diffusion equations it is very interesting to note that in all differential equations dimensionless numbers are very relevant. These equations are no exception to this fact, in this case the most relevant dimensionless number is the Peclet number (Pe). It is defined as:

$$Pe = \frac{\rho u L}{\Gamma}$$

This number is extremely relevant for the analytical solution of these equations (which can be obtained for a very simplified one dimensional case) and of course for the numerical solution as well. This number is responsible for the divergence of the solution in some numerical schemes as will be explained in the results section of this problem.

## Problem solution

Like in the previous problem the goal is to simplify the solution into a more compact notation with known coefficients:

$$a_P \phi_P = a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S + b_n$$

In this case a low order implicit scheme will be used.

The original equation that was obtained is the solution for a Central Difference Scheme (CDS), which is a second order scheme that supposes that the value of a property in the cell face is the arithmetic mean of the properties at the nodes:

$$\phi_e = \frac{1}{2}(\phi_P + \phi_E)$$

This is consistent with a second order Taylor series development:

$$\phi(x) = \phi(x_0) + \frac{d\phi}{dx} * (x - x_0) + \frac{d^2\phi}{dx^2} * (x - x_0)^2$$

On the other hand an Upwind Difference Scheme (UDS) can also be used. It supposes that the value at the cell face is the equal to the value at the node that is upwind (that a particle would have left behind previous to reaching the cell face). This scheme tries to better

anticipate the changes that will happen to the fluid due to convective effect, but it is only a first order scheme and therefore the solution should have a higher error value than in second order schemes:

$$\begin{aligned}\phi_e &= \phi_P \text{ if } F_e > 0 \\ \phi_e &= \phi_E \text{ if } F_e < 0\end{aligned}$$

Also a hybrid scheme can be used. Hybrid schemes are a mix of CDS and UDS and apply a CDS scheme for low velocities and an UDS scheme for high velocities.

An Exponential Difference Scheme can also be used. This scheme is considered because it corresponds to the exact solution of the convection-diffusion equation. This solution has not been used in this work, but it is possible to solve the convection-diffusion equations analytically in the case of one dimensional null source term and steady solution. This yields the following exponential solution, which is also a second order approximation:

$$\frac{\phi - \phi_0}{\phi_L - \phi_0} = \frac{e^{\frac{Px}{L}} - 1}{e^P - 1}, \quad (\phi_0 = \phi(x = 0), \phi_L = \phi(x = L))$$

Finally a Powerlaw Difference Scheme (PLDS) is a second order approximation that is an approximation of the analytical exponential solution of the problem but uses a fifth degree polynomial solution instead.

The solution of the problem now comes from the reorganization of the discretised convection-diffusion equation. Our goal is to obtain the different values of the coefficients of the simplified equation for each numerical scheme. The best way to do so is using the coefficient A(P), which shall be dependent on the numerical scheme and will introduce the dependence of the problem on the Peclet number, the coefficient D and F, which is the flux through the cell face. Then the coefficients become:

$$a_E = D_e * A(|Pe|) + \max(-F_e, 0)$$

$$a_S = D_s * A(|Pe|) + \max(-F_s, 0)$$

$$a_N = D_n * A(|Pe|) + \max(-F_n, 0)$$

$$a_S = D_s * A(|Pe|) + \max(-F_s, 0)$$

$$a_p = a_E + a_W + a_N + a_S + \frac{\rho_p^n \Delta x \Delta y}{\Delta t}$$

$$b_P = \frac{\rho_p^n \Delta x \Delta y}{\Delta t} \phi_p^n + S_p^{n+1} \Delta x \Delta y$$

Where:



$$D_e = \frac{\Gamma \Delta y}{dx_e}, \quad D_w = \frac{\Gamma \Delta y}{dx_w}$$

$$D_n = \frac{\Gamma \Delta y}{dx_n}, \quad D_s = \frac{\Gamma \Delta y}{dx_s}$$

$$F_e = (\rho u)_e \Delta y, \quad F_w = (\rho u)_w \Delta y$$

$$F_n = (\rho u)_n \Delta y, \quad F_s = (\rho u)_s \Delta y$$

And the Peclet number is:

$$P_f = \frac{F_f}{D_f}$$

The solution for each case is presented in the following table:

Numerical Scheme	$A( Pe )$
<b>UDS</b>	1
<b>CDS</b>	$1 - 0.5( Pe )$
<b>HDS</b>	$\max(0, (1 - 0.5( Pe )))$
<b>EDS</b>	$ Pe /(e^{ Pe } - 1)$
<b>PLDS</b>	$\max(0, (1 - 0.5( Pe )^5))$

Table 4: Value of  $A(|Pe|)$  for low numerical schemes

### Numerical Solution

The implementation of the code is similar to the base exercise solution. In this case the nodes are rectangular, the domain is divided in  $nx$  control volumes in the x direction and in  $ny$  control volumes in the y direction. Since an additional node is added in the boundary the number of nodes shall be  $nx+2$  and  $ny+2$  respectively.

The function defcoord determines the position of each node and stores it, this will be very useful to compute the velocity field. This is done in the function defvel. Finally the initial temperature map is established in the function defT.

## Coefficient determination

The main difficulty of the program implementation is to compute the coefficients at each control volume. On the one hand the code has to be implemented in a manner in which the user can define the numerical scheme (variable method). Therefore all the numerical schemes have to be implemented. On the other hand special attention has to be dedicated to boundary nodes, since the equations that rule the general nodes cannot be applied directly.

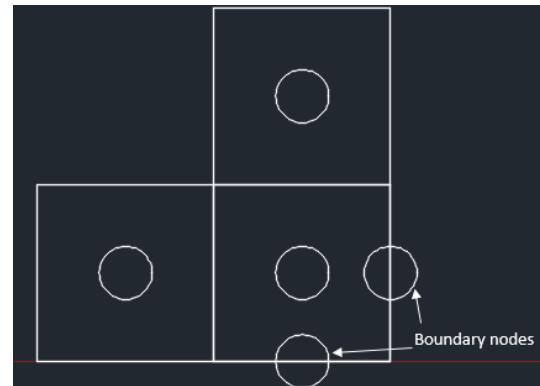


Figure 21: Boundary nodes

Also the nodes that are next to the boundary nodes have a different expression due to the proximity of these nodes. Therefore for boundary nodes and the adjacent nodes it is important to take into account that the distance to the next node is  $\text{deltax}/2$  instead of  $\text{deltax}$  which applies to any other node.

Another modification that has to be made when computing the coefficients is to take into account the boundary conditions. This again affects the boundary nodes. There are two main boundary conditions; for the left, top, right and low-left boundaries the temperature is determined by the problem. The first three boundaries correspond to a wall and the third corresponds to the inlet. This corresponds to a value of 0 for all the coefficients except  $a_p$  ( $a_E = a_W = a_N = a_S = b_p = 0$ ). At Figure 22 this corresponds to the boundaries that are coloured in blue.

The low-right boundary ( $y = 0$  and  $x > 0$ ) is the outlet and to compute its temperature the best solution is to ignore the convective and conductive term in the boundary nodes (that have no volume) with the other boundary nodes. By doing this the boundary coefficients become 0 except for  $a_p$  and  $a_N$  and therefore the temperature at the outlet will be exactly the same as the temperature at the node which is directly next to it vertically.  $a_E = a_W = a_S = b_p = 0$ . At Figure 22 this corresponds to the boundaries that are coloured in red.

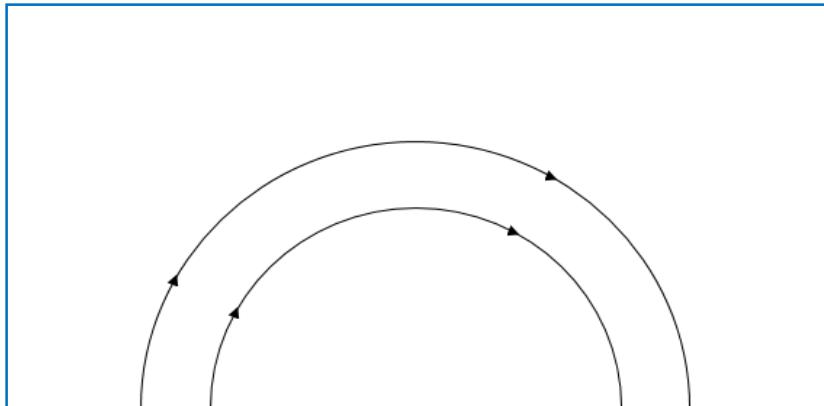


Figure 22: Boundary properties

### Main program

The main program establishes the main geometrical and physical properties first and then determines the solution of the problem. To do this, first the different possible values of the main parameter of the program ( $\rho/\Gamma$ ) are established. The three possible values are:

$\rho/\Gamma$		
10	$10^3$	$10^6$

Table 5: Possible values of  $\Gamma/\rho$

This parameter is extremely relevant for the problem solution because it will determine the relationship between the convective and the conductive term. By recalling the conduction-diffusion equation this becomes obvious:

$$\frac{d(\rho\phi)}{dt} + \nabla(\rho v\phi) = \nabla(\Gamma\nabla\phi)$$

But first we must assume that  $\Gamma$  and  $\rho$  are constant (which is nearly compulsory if the relationship between them is fixed). Then:

$$\frac{d(\phi)}{dt} + \nabla(v\phi) = \frac{\Gamma}{\rho} \nabla(\nabla\phi)$$

By studying the equation it becomes also obvious that high values of  $\rho/\Gamma$  imply that the convective term shall be the most relevant term and that low values  $\rho/\Gamma$  imply that the conductive term shall be the most relevant term.

After determining the initial velocity and temperature fields and imposing the geometry of the problem with the functions defvel, defT and defcoord that have already been mentioned the iterative process is started. When starting a new time step the coefficients for this time step are determined. Then, the temperature is determined with the iterative process and then for each node the convergence condition is checked. When in all nodes the temperature change is lower than the allowed error, the next time step is computed

with the same method. When all the time steps have been computed the results are printed in .dat files with similar routines than in the base exercise:

#### Program output functions:

To compare the program solution with the provided solution it is required to compute the temperature at the outlet. When trying to use the routine developed for this purpose in the base exercise, a problem appeared: the routine tried to use the temperature of the nodes that were below the boundary (and do not exist). To avoid this problem a new case was developed: instead of using a four node interpolation, now the program uses a two node interpolation between the closest two boundary nodes.

$$T_P = T_n + \frac{(x_P - x_n)}{x_{n+1} - x_n} T_{n+1}$$

The function entrega is similar to the entrega function from the base program. In this case it writes a .dat file containing the temperature values for the nodes corresponding to the provided numerical solution. The function treuredades prints the temperature of every node in a .dat file as well as its coordinates in a fashion that the program gnuplot can use to plot the temperature distribution graphically:

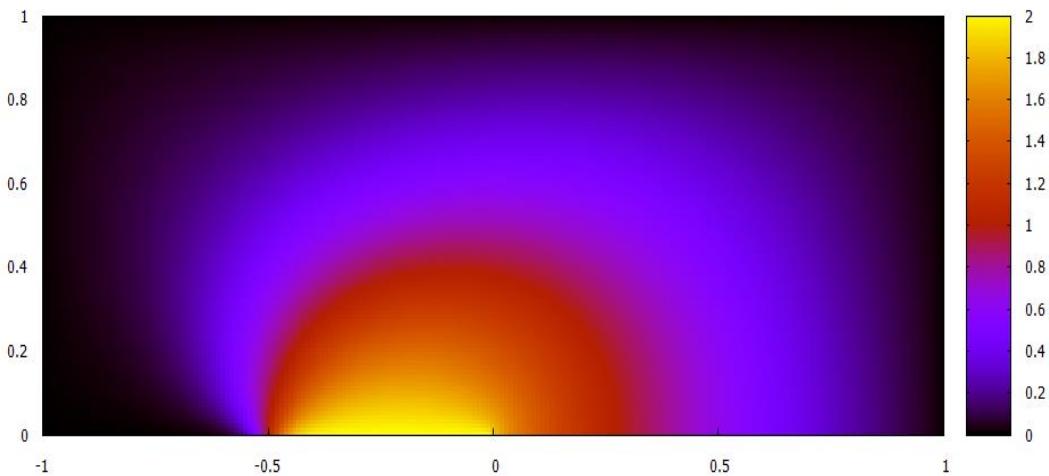


Figure 23: Temperature distribution for  $\frac{\Gamma}{\rho} = 10$

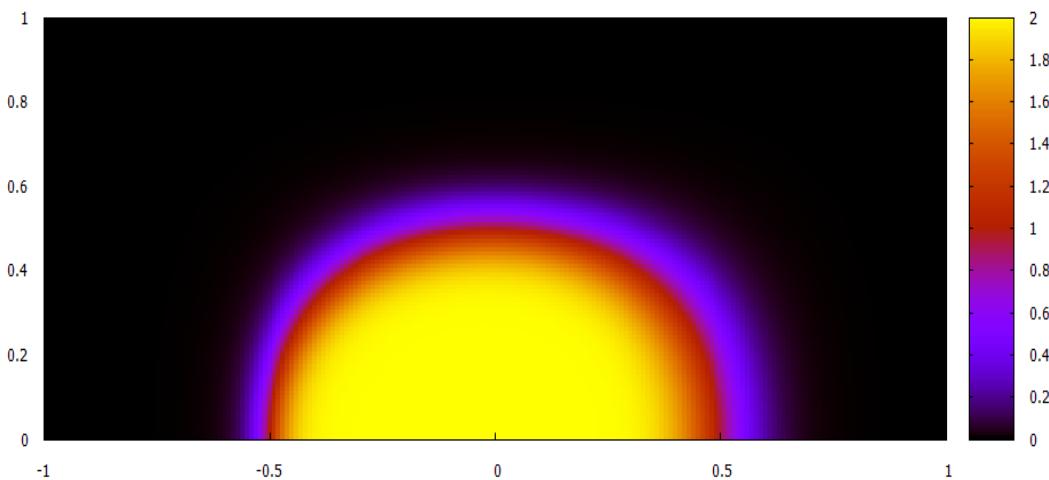


Figure 24: Temperature distribution for  $\frac{\Gamma}{\rho} = 1000$

### General program scheme

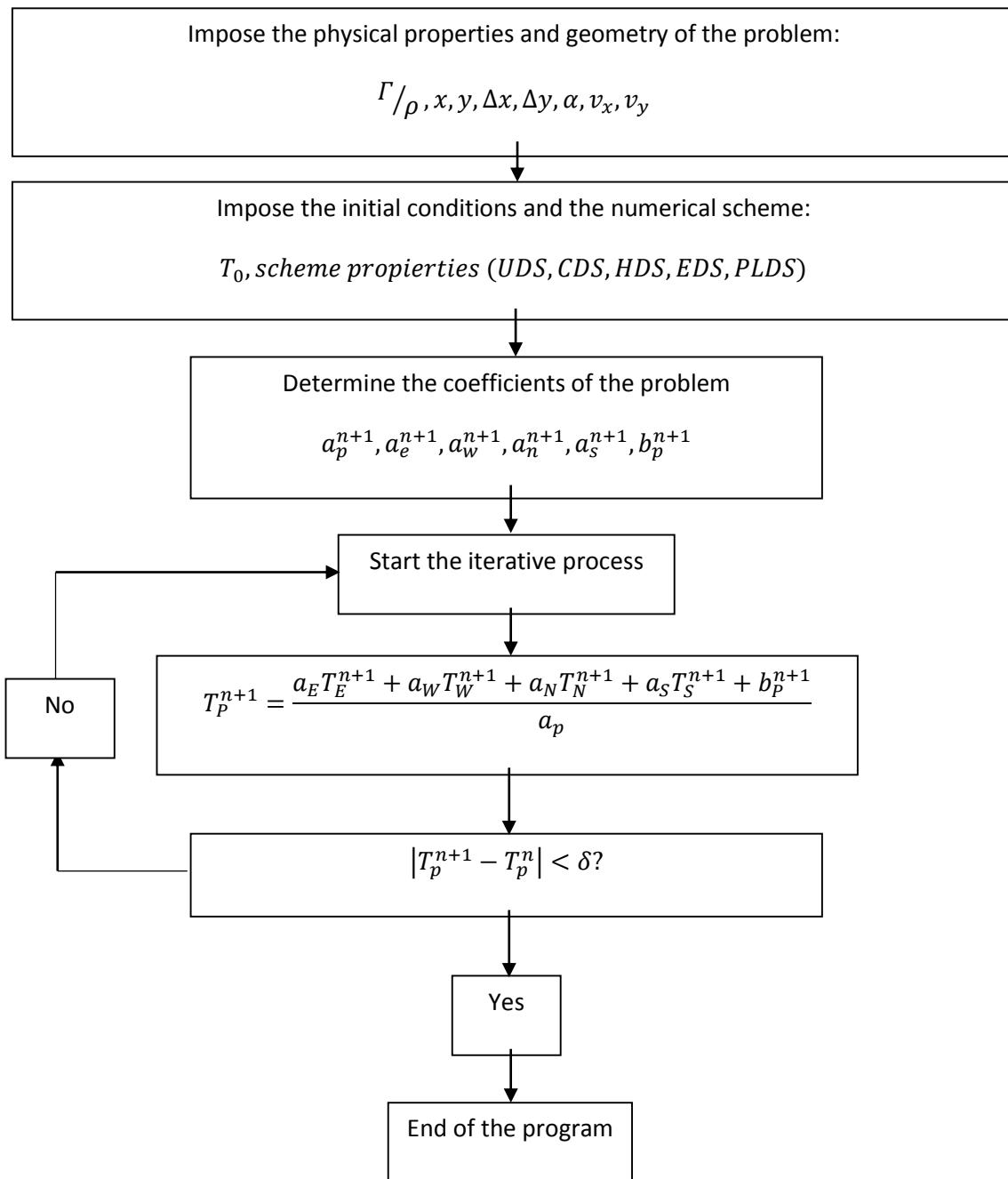
This program can use five different numerical schemes to solve the conduction equations. Note that the use of numerical schemes can never affect the solution of the problem, it shall only affect the convergence conditions (which time steps and separation between nodes are required for each value of  $\frac{\Gamma}{\rho}$ ). The solution of the system of equations will then be:

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

In this problem the general scheme of the program is preserved for all the different numerical schemes, since the development is only valid for an implicit solution.

Also, since the velocity field is imposed at all points of the domain there will be no difference when solving the iterative process to the iterative process of the base exercise.

The general scheme of the program will then be very similar to the general scheme of the base exercise, except for the fact that in this case we are only interested in the steady state solution of the problem and therefore no temporal evolution of the problem will be required. The program will directly iterate the equations to find the steady state solution and when the solution is found there will be no more time steps to iterate.



## Results

In this section the results obtained by the different numerical schemes will be compared. It is important to bear in mind that ideally the results should be identical. Unfortunately some schemes present difficulties in the form of non-convergence. Specifically when trying to solve the problem with the CDS a non-convergence error has appeared. This problem shall be explained below and is related to the high values that the Peclet number can reach.

To study this problem it is important to understand the inlet conditions. These conditions are defined by an hyperbolic tangent which is not a function that is easily visualised. From the problem definition:

$$\phi = 1 + \tanh(\alpha(2x + 1)) \text{ at the inlet}$$

So, since this function is complex the best way to understand it is to plot it with Matlab (see Figure 25):

This function has the peculiarity of being very close to 0 from  $x = -1$  to  $x = -0.6$  and very close to 2 from  $x = -0.4$  to  $x = 0$ . In between these two points the temperature derivative is very high. So, the inlet conditions determine two separate regions: near the centre of the cavity the temperature is nearly 2 and near the wall the temperature is nearly 0. Now the subject of study is to determine how this hyperbolic tangent distribution is altered taking into account the properties of the fluid:

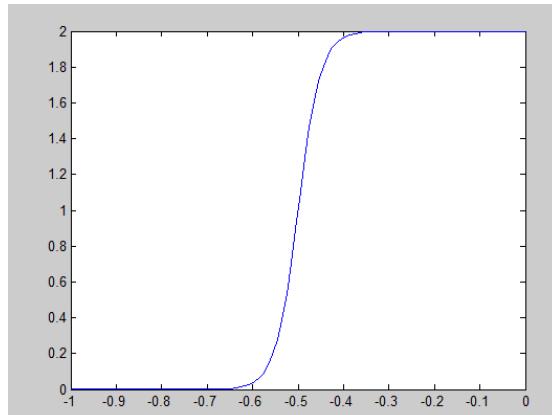


Figure 25: Hyperbolic tangent representation

When studying the results it is expected that for high values of  $\Gamma/\rho$  the convective term shall be predominant (see the problem solution for a more extensive mathematical explanation) whilst for low values of this term the conductive term shall dominate. The physical meaning of this is simple, for high values of  $\Gamma/\rho$  the flux moves “too fast” for conduction to affect the temperature field. Then the flux that is entering the cavity circulates through it without exchanging an important quantity of heat during this period. This is translated in a much separated flux that in Figure 26 is nearly unaltered at the outlet. The flux properties at the outlet shall be nearly identical to the inlet flux properties:

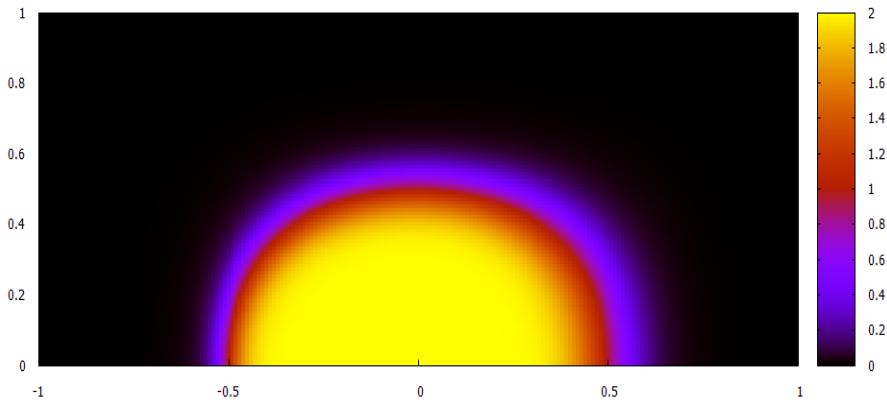


Figure 26: Temperature distribution for  $\frac{\Gamma}{\rho} = 1000$

If a contour plot is drawn the lines that follow the constant temperature will be nearly circular, since a particle that has entered the inlet at a temperature leaves the outlet at nearly the same temperature. Therefore the constant temperature lines shall resemble the path of the incoming particles. As expected in Figure 27 the contour lines are only slightly modified from the circular path of the flux.

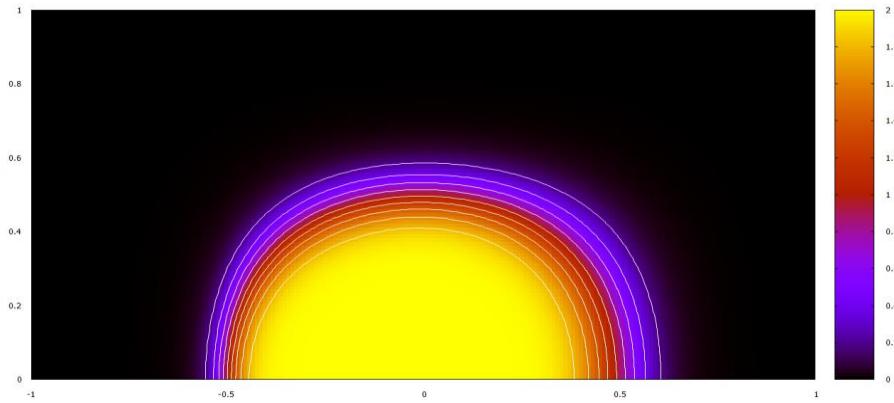


Figure 27: Contour plot of the temperature distribution for  $\frac{\Gamma}{\rho} = 1000$

The same plots can be obtained for higher values of the  $\Gamma/\rho$  parameter, but the temperature field is not very different from the temperature field obtained with  $\Gamma/\rho = 1000$ . The convective term is even more important than in the previous case

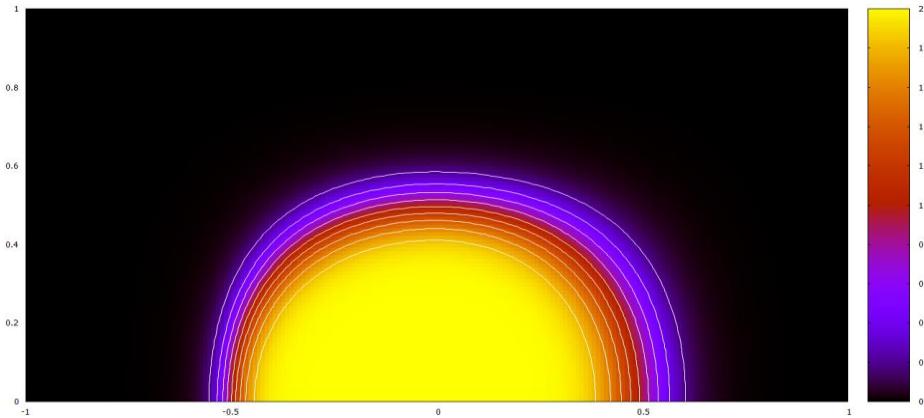


Figure 28: Contour plot of the temperature distribution for  $\frac{\Gamma}{\rho} = 10^6$

On the other hand low values of  $\frac{\Gamma}{\rho}$  shall mean that the conductive term is dominant. This is equivalent to saying that the flux spends “a long time” in the cavity during which the particles diffuse and the temperature field at the inlet is modified by this diffusion. Diffusion tends to equilibrate the temperature of the fluid and reduce the temperature gradients, so in the region in which the temperature gradient is high ( $r=0.5$ ) there will be a very active heat exchange process. From Figure 29 it is clear that the diffusive process has altered the temperature distribution and that the outlet conditions do not resemble the inlet conditions. Energy must be conserved and therefore the heat flows from the centre of the cavity to the exterior.

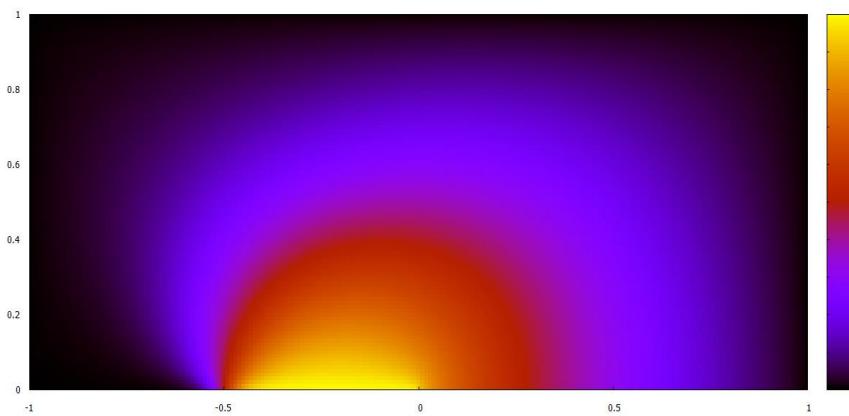


Figure 29: Temperature distribution for  $\frac{\Gamma}{\rho} = 10$

If a contour plot is drawn the lines that follow the constant temperature shall clearly differ from the circular path of the flux. The centre of such lines is no longer the centre of the cavity and the contour lines resemble more a spiral than a circle. The maximum value of the temperature can only be present in the centre because it is next to the inlet and the outlet has a rather smooth temperature distribution. The diffusive term is responsible for this evolution of the temperature field, which is extremely clear in Figure 30:

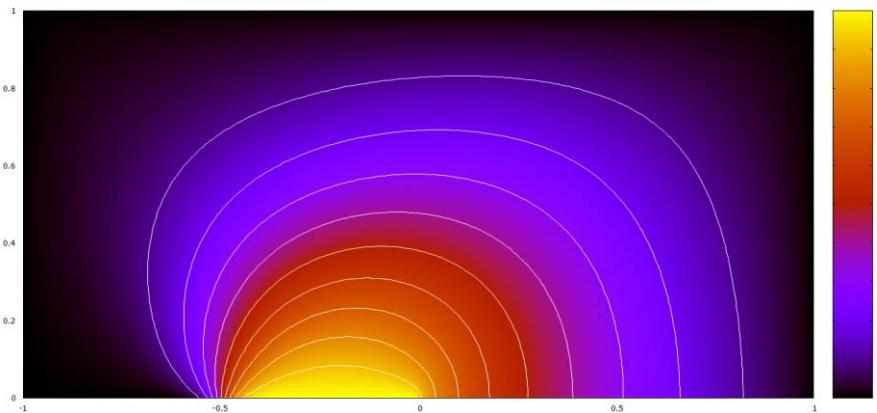


Figure 30: Contour plot of the temperature distribution for  $\frac{\rho}{\Gamma} = 10$

Having solved the problem now it is interesting to compare the different numerical schemes.

When comparing the results it is impossible to compare the whole temperature map. For this reason only the outlet conditions are studied. The data obtained with the different schemes is presented in the corresponding annex. To better compare the different values the data is obtained with the same mesh and time step, the convergence criteria is the same and therefore the only difference is the numerical scheme. Then the results can be plotted in Matlab (see Figure 31, Figure 32, Figure 33) for the three cases that have already commented.

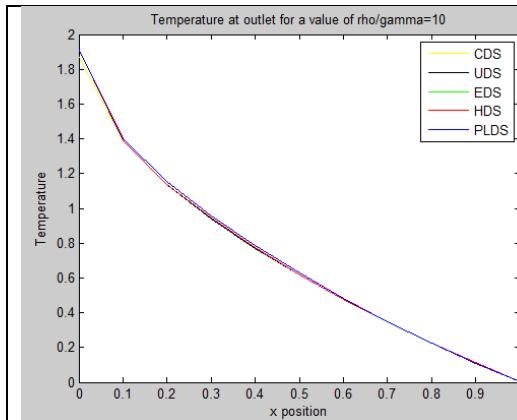


Figure 31: Temperature at outlet for  $\rho/\Gamma = 10$

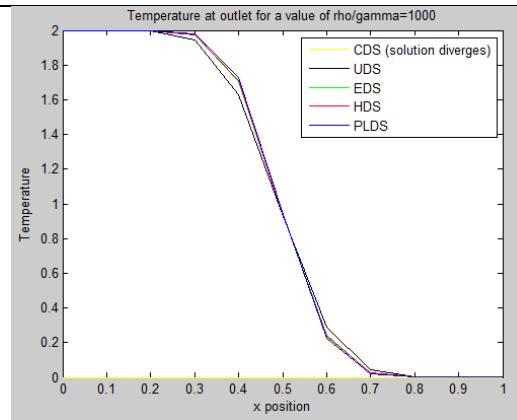


Figure 32: Temperature at outlet for  $\rho/\Gamma = 10^3$

For  $\rho/\Gamma = 10$  the temperature distribution for the different numerical schemes is nearly identical as expected. In this case none of the methods can be selected or discarded with this information.

For  $\rho/\Gamma = 10^3$  the temperature distribution for the different numerical schemes is nearly identical as expected except for CDS. This is due to the construction of the CDS method. For CDS:

$$A_e = 1 - 0.5(|Pe|)$$

This means that for high Peclet values ( $Pe > 10$ )  $A_e$  can also reach very high values. Peclet is computed with the following formula:

$$Pe = \frac{\rho u L}{\Gamma}$$

It is clear that if the factor  $\rho/\Gamma$  increases then the Peclet value will increase accordingly. For  $\rho/\Gamma = 10$ , the Peclet value had a moderate value and therefore  $A_e$  had also a moderate value. Then the CDS method could converge. In this case, instead, the factor  $\rho/\Gamma$  is increased in a factor of 100. Then the Peclet value is also increased in a factor of 100 and therefore the convergence of the method is compromised. This fact can

be compensated by increasing the mesh density, but at the cost of a much higher computational cost. Therefore the CDS method is not recommended when trying to solve this problem. The other schemes offer very similar results and therefore are equivalent from the point of view of reaching the correct solution.

For these conditions ( $\rho/\Gamma = 10^6$ ) the temperature distribution for the different numerical schemes is nearly identical as expected. In this case none of the methods can be selected or discarded with this information, since the CDS method was already discarded in the previous case.

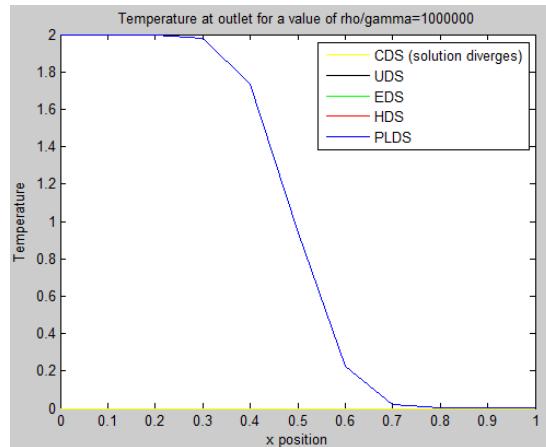


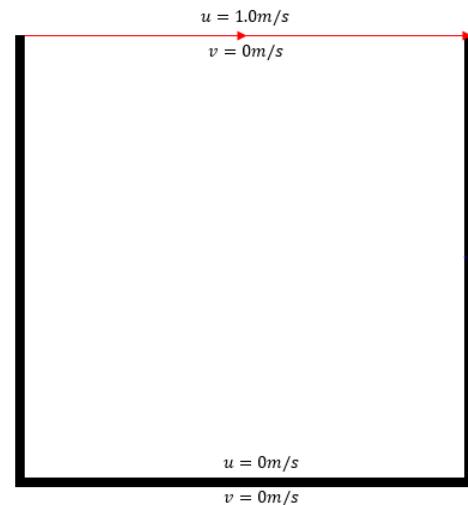
Figure 33: Temperature at outlet for  $\rho/\Gamma = 10^6$

# The driven cavity problem

## Introduction

The main goal of this exercise is to solve the following bi-dimensional incompressible laminar problem:

The problem geometry consists in a square cavity that lacks the top wall. In this region ( $y=1m$ ) the velocity is imposed as a horizontal  $1m/s$  speed, and on the walls the speed is 0 (non-slip condition). The goal of the problem is to determine the speed distribution of the cavity. To do this the laminar Navier-Stokes equations must be solved. This is a commonly used problem as a model for testing and evaluating numerical techniques.



This problem shall be solved for a wide range of [Figure 34: Driven cavity problem](#) Reynolds that go from low values to moderate values (higher values should take into account the effect of turbulence).

The Navier Stokes equations are:

$$\frac{du}{dt} + (\mathbf{u} \cdot \nabla) \mathbf{u} = \frac{1}{Re} \Delta \mathbf{u} - \nabla p$$

$$\nabla \cdot \mathbf{u} = 0$$

Where Re is the Reynolds dimensionless number:

$$Re = \frac{\rho V_0 L}{\mu}$$

## Application of the Helmholtz-Hodge decomposition theorem

Using the Helmholtz-Hodge decomposition theorem it can be demonstrated that the Navier-Stokes equations can be split in two parts: a divergence free vector and a gradient of a scalar field.

In this case the divergence free vector shall include the convective and diffusive terms, whilst the scalar field shall be the pressure. These two terms can be separately obtained by applying the projector operator ( $\Pi(\cdot)$ ) to the differential Navier-Stokes equations:

$$\Pi \left( \frac{du}{dt} + \nabla p \right) = \Pi \left( -(\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{Re} \Delta \mathbf{u} \right)$$

$$\nabla \prod(\mathbf{a}) = 0$$

The most interesting property of the projector operator is that it projects a vector onto a divergence free space. This has three main consequences:

- 1- A divergence free vector will remain unchanged.
- 2- The projection of a gradient field shall always be 0.
- 3- This decomposition is unique

This can be better understood with the help of the following diagram provided by the CTTC theoretical development<sup>[2]</sup>:

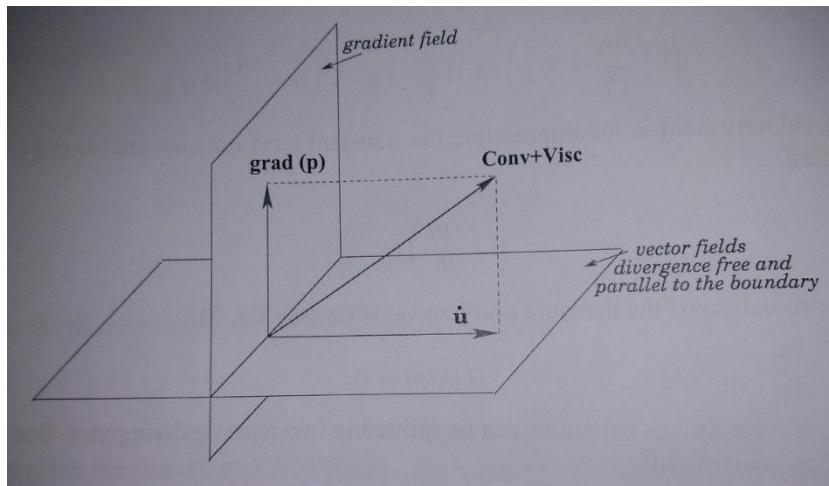


Figure 35: Convective + viscous vector field decomposition.

Then the application of this theorem has the following consequences on the Navier-Stokes equations terms:

**Temporal derivative:**

Since the temporal derivative of an incompressible field is also incompressible, then it shall also be a divergence free vector. Therefore the application of the projector operator shall not affect this term:

$$\prod\left(\frac{du}{dt}\right) = \frac{du}{dt}$$

**Pressure gradient:**

The pressure gradient is a gradient field and according to the already mentioned consequences its projection shall always be 0.

$$\prod(\nabla p) = 0$$



Convective-diffusive term:

This term cannot be simplified by the projector operator.

$$\prod \left( -(\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{Re} \Delta \mathbf{u} \right)$$

We will name:

$$\mathbf{R}(\mathbf{u}) = -(\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{Re} \Delta \mathbf{u}$$

Navier-Stokes split equations

This leads to the following two Navier-Stokes equations:

$$\frac{d\mathbf{u}}{dt} = \prod \left( -(\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{Re} \Delta \mathbf{u} \right) \text{(first equation)}$$

$$\nabla p = \frac{1}{Re} \Delta \mathbf{u} - (\mathbf{u} \cdot \nabla) \mathbf{u} - \prod \left( -(\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{Re} \Delta \mathbf{u} \right) \text{(second equation)}$$

Then applying the divergence operator and using the projector definition it is possible to obtain:

$$\Delta p = \nabla \cdot \left( -(\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{Re} \Delta \mathbf{u} \right)$$

It is possible then to conclude that for incompressible flows the pressure gradient plays the role of projecting the convective-diffusive term ( $\mathbf{R}(\mathbf{u})$ ) into a divergence free space.

### Discretisation of the equations

Then the problem shall be to solve the following equation:

$$\frac{d\mathbf{u}}{dt} = \mathbf{R}(\mathbf{u}) - \nabla p$$

Recall that

$$\mathbf{R}(\mathbf{u}) = -(\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{Re} \Delta \mathbf{u}$$

This problem can be discretized in many fashions, but to reduce the amount of calculations and for the sake of simplicity and clarity a fully explicit time integration scheme will be used. A second order Adams-Bashford scheme will be used for  $\mathbf{R}(\mathbf{u})$ . This leads to the following discretised equation:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \frac{3}{2} \mathbf{R}(\mathbf{u}^n) - \mathbf{R}(\mathbf{u}^{n-1}) - \nabla p^{n+1}$$



Then to split this equation the Helmholtz-Hodge decomposition will be used. As it has already explained before the velocity can be split in two terms: a predictor velocity and the pressure gradient.

$$\mathbf{u}^{n+1} = \mathbf{u}^p - \nabla p$$

Then to obtain  $\mathbf{u}^{n+1}$  one must first obtain the predictor velocity and then compute the pressure gradient to ensure that  $\mathbf{u}^{n+1}$  is incompressible.

Then, from the first equation:

$$\mathbf{u}^p = \mathbf{u}^n + \Delta t \left( \frac{3}{2} \mathbf{R}(\mathbf{u}^n) - \frac{1}{2} \mathbf{R}(\mathbf{u}^{n-1}) \right)$$

Then one can identify a pseudo-pressure that shall be:  $\tilde{p} = \Delta t p^{n+1}$ . This leads to the so-called Poisson-equation which can be obtained from the second equation:

$$\Delta \tilde{p} = \nabla \mathbf{u}^p$$

Finally the velocity at the next time step can be obtained:

$$\mathbf{u}^{n+1} = \mathbf{u}^p - \nabla p$$

Then the equations must be discretised so they can be solved numerically. To do this, the equations will be discretised in the order that they have to be solved to correctly compute the velocity at the next time step:

#### 1- Discretisation of $\mathbf{R}(\mathbf{u})$ :

Applying the gauss theorem one can easily transform a derivative equation into an integral equation, which can then be discretised:

$$\int_V \mathbf{R}(\mathbf{u}) dV = - \int_V (\mathbf{u} \cdot \nabla) \mathbf{u} dV + \frac{1}{Re} \int_V \Delta \mathbf{u} dV$$

To simplify this instead of discretising  $\mathbf{u}$ , a generic variable  $\phi$  shall be discretised in each direction (x,y):

$$\mathbf{R}_x(\phi) dx dy = - \int_V \mathbf{u} \frac{d\phi}{dx} dV + \frac{1}{Re} \int_V \frac{d^2 \phi}{dx^2} dV$$

$$\mathbf{R}_y(\phi) dx dy = - \int_V \mathbf{v} \frac{d\phi}{dy} dV + \frac{1}{Re} \int_V \frac{d^2 \phi}{dy^2} dV$$

Then:

$$\mathbf{R}_x(\phi) dx dy = - \int_S \mathbf{u} \phi dS + \frac{1}{Re} \int_S \frac{d\phi}{dx} dS$$



$$\mathbf{R}_y(\phi) dx dy = - \int_S \mathbf{v} \phi dS + \frac{1}{Re} \int_S \frac{d\phi}{dy} dS$$

Finally:

$$\begin{aligned} \mathbf{R}_x(\phi) dx dy &= (\mathbf{u}\phi)_e S_e - (\mathbf{u}\phi)_w S_w + (\mathbf{u}\phi)_n S_n - (\mathbf{u}\phi)_s S_s \\ &\quad + \frac{1}{Re} \left[ \frac{d\phi}{dx} \Big|_e S_e - \frac{d\phi}{dx} \Big|_w S_w + \frac{d\phi}{dx} \Big|_n S_n - \frac{d\phi}{dx} \Big|_s S_s \right] \end{aligned}$$

$$\begin{aligned} \mathbf{R}_y(\phi) dx dy &= (\mathbf{v}\phi)_e S_e - (\mathbf{v}\phi)_w S_w + (\mathbf{v}\phi)_n S_n - (\mathbf{v}\phi)_s S_s \\ &\quad + \frac{1}{Re} \left[ \frac{d\phi}{dx} \Big|_e S_e - \frac{d\phi}{dx} \Big|_w S_w + \frac{d\phi}{dx} \Big|_n S_n - \frac{d\phi}{dx} \Big|_s S_s \right] \end{aligned}$$

2- Discretisation of  $\mathbf{u}^p$ :

$$\mathbf{u}^p = \mathbf{u}^n + \Delta t \left( \frac{3}{2} \mathbf{R}(\mathbf{u}^n) - \frac{1}{2} \mathbf{R}(\mathbf{u}^{n-1}) \right)$$

This equation can be applied directly

3- Discretisation of the Poisson equation

$$\begin{aligned} \int_V \Delta \tilde{p} dV &= \int_V \nabla \mathbf{u}^p dV \\ \int_S \nabla \tilde{p} dS &= \int_S \mathbf{u}^p dS \\ \frac{P_E - P_P}{X_E - X_P} S_E + \frac{P_W - P_P}{X_W - X_P} S_W + \frac{P_N - P_P}{X_N - X_P} S_N + \frac{P_S - P_P}{X_S - X_P} S_S \\ &= (\mathbf{u}^p S)_e - (\mathbf{u}^p S)_w + (\mathbf{u}^p S)_n - (\mathbf{u}^p S)_s \end{aligned}$$

Then the equation can be simplified to:

$$a_P P_P = a_E P_E + a_W P_W + a_N P_N + a_S P_S + b_P$$

Where  $a_P, a_E, a_W, a_N$  and  $a_S$  are constant through the whole program at each node, while  $b_P$  is dependent on  $\mathbf{u}^p$  and will have to be computed at each iteration.

This system can be solved iteratively with a Gauss-Seidel method.

4- Discretisation of  $\mathbf{u}^{n+1}$

$$\int_V \mathbf{u}^{n+1} dV = \int_V \mathbf{u}^p dV - \int_V \nabla p dV$$

$$\int_V \mathbf{u}^{n+1} dV = \int_V \mathbf{u}^p dV - \int_V \nabla p dV$$

$$\mathbf{u}^{n+1} dx dy = \mathbf{u}^p dx dy - \int_S p dS$$

Finally:

$$\mathbf{u}^{n+1} = \mathbf{u}^p - \frac{[(pS)_e + (pS)_w + (pS)_n + (pS)_s]}{dxdy}$$

## Numerical solution

When solving this problem numerically there are two important issues to address: first, the mesh construction, and second the program construction.

### Solution using a staggered mesh

When solving velocity field problems it might be interesting to take into account using a staggered mesh. Staggered meshes consist in using different meshes for the physical properties of the problem (pressure, temperature...) than for the velocity. Why is this interesting?

When solving this problem it is clear that the equations require the flux and the velocity at the control volume boundary. A first approach when trying to compute it is to determine the arithmetic mean of the two closest nodes. Staggered meshes take another approach and directly place the velocity at the control volume boundary. This has the beneficial consequence of simplifying the calculations because the velocity is already placed in the correct position in the majority of the equations. But it also adds an additional difficulty, in some other equations it is necessary to compute the velocity at the staggered control volume boundary, and therefore it is not an enormous gain. This difference is clearer in Figure 36.

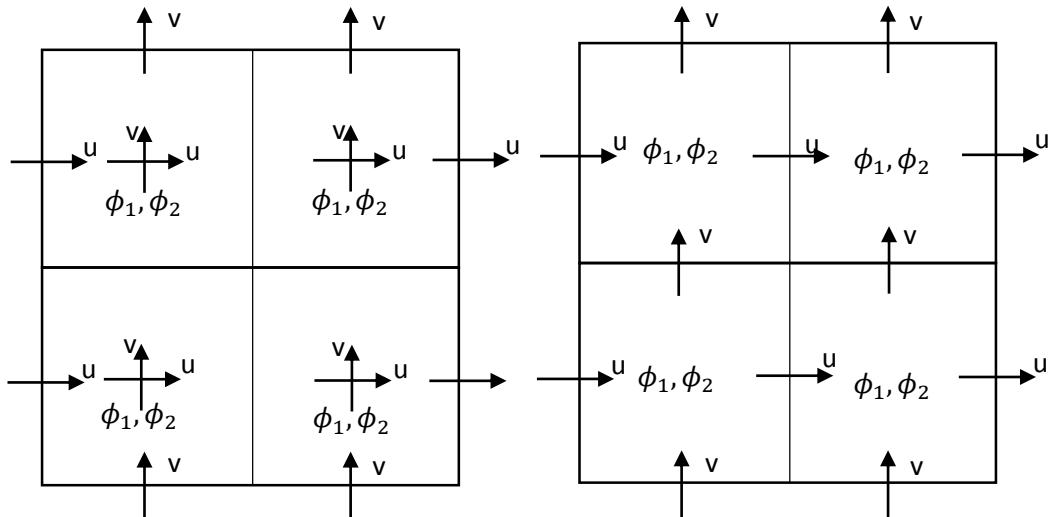


Figure 36: Face centred and staggered meshes

Note that this is a 2 node discretisation of the domain, an additional node is added to the domain boundary to better impose the boundary conditions, although it is not absolutely

necessary. With the help of Figure 36 it is clear that the staggered mesh requires one less node in the direction of the velocity (for example  $\mathbf{u}$  is a  $3 \times 4$  matrix in the staggered mesh and a  $4 \times 4$  matrix in the face centred mesh). The same can be done with the fluxes since  $\mathbf{F} = \rho \mathbf{v}$ . If the velocity field mesh is considered, two staggered meshes appear:

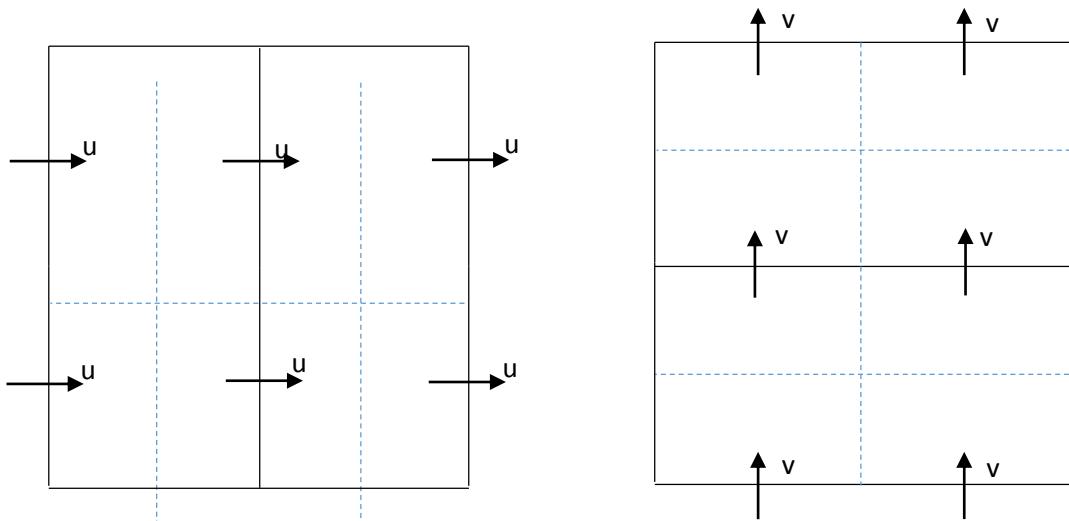


Figure 37:  $x$  and  $y$  velocity staggered meshes

When solving the following equation some problems appear:

$$\begin{aligned} \mathbf{R}_x(\phi) dx dy &= (\mathbf{u}\phi)_e S_e - (\mathbf{u}\phi)_w S_w + (\mathbf{u}\phi)_n S_n - (\mathbf{u}\phi)_s S_s \\ &\quad + \frac{1}{Re} \left[ \frac{d\phi}{dx} \Big|_e S_e - \frac{d\phi}{dx} \Big|_w S_w + \frac{d\phi}{dx} \Big|_n S_n - \frac{d\phi}{dx} \Big|_s S_s \right] \end{aligned}$$

- 1- When trying to compute the terms  $(\mathbf{u}\phi)_e S_e = \mathbf{F}_e * \phi_e = \mathbf{F}_e * u_e$  the velocity is required at the east face of the control volume. It can be computed with an arithmetic mean:  $u_e = \frac{u_E + u_P}{2}$  and  $F_e = u_e S_e = u_e dy$ . The same can be applied to  $u_w$
- 2- When trying to compute the terms  $(\mathbf{u}\phi)_n S_n = \mathbf{F}_n * \phi_n = \mathbf{F}_n * u_n$  the velocity is required at the north face of the control volume. It can be computed with an arithmetic mean:  $u_n = \frac{u_N + u_P}{2}$  and  $F_n = v_n S_n = v_n dx$ . But  $v_n$  does not coincide with any velocity that is available since the other staggered mesh positions do not have a node in this point. The velocity which is required is right in the middle of two points and therefore it can be computed as the arithmetic mean of these two points. Then  $v_n = \frac{v_{ne} + v_{nw}}{2}$ . Where  $v_{ne}$  and  $v_{nw}$  are the two closest points that lie north of the control volume. The same can be applied to  $u_s$ .



- 3- To compute the derivatives of the velocity a second order Taylor series development is used:

$$\frac{du}{dx}\Big|_e = \frac{u_E - u_P}{dx}$$

This can be applied in any direction.

The other three equations do not require any special attention since all the required terms have already been computed and are placed in the correct position.

## Meshes

The selection of an adequate mesh is very important in this problem, especially for high Reynolds numbers. A first approach to this problem was to choose a square homogeneous mesh, very similar to the meshes used in the previous exercises. To increase the precision of the solution a second approach consisted in developing a structured mesh.

### Structured meshes

Structured meshes try to increase the precision of the solution by concentrating a higher number of nodes in those positions where the velocity gradients are high and separating the nodes in the positions where the velocity gradients are low. Obviously the nodes should be more concentrated near the walls (to better define the boundary layer) and more separated at the centre.

Typical functions that have a value distribution which is relatively constant at the centre and much smaller at the sides are the hyperbolic relationships. In this case a hyperbolic tangent function will be used to describe the position of the nodes. This function has been provided by professor Xavier Trias from the CTTC<sup>[4]</sup>. This function describes the position of the boundaries:

$$x_b = \frac{X_{max}}{2} * \left( 1 + \frac{\tanh\left(\gamma \left( \frac{2(i-1)}{N} \right) - 1\right)}{\tanh \gamma} \right)$$

Where  $X_{max}$  is the dimension of the driven cavity,  $\gamma$  is the concentration factor (0 corresponds to an homogeneous mesh) and N is the number of nodes.

## Program functioning

When started, the program first sets the basic constants of the problem such as the Reynolds number, the basic dimensions, and the basic structures of the problem.

## Program structures

The program utilises structures to better store the information in a fashion that is more comprehensible and easier to work with:



The structures posx and posy have already been used in prior programs and store the x and y position of the nodes, and their boundaries (e,w,n,s).

The structure PECLET stores the value of a generic property in all the positions of a node. Therefore when computing for example the pressure the program can store the value of P at each node and also its value at the boundaries (e,w,s,n). With this method the most relevant properties of the problem are stored: pressure, velocity and flux. Note that the velocity and flux require one node less in one of the dimensions, the extra node is left blank and fixed at 0. Also note that the name PECLET has no relationship with the actual Peclet number, in this case it only functions as a tag.

### Initial conditions

At the program start the velocity and pressure maps are known: the pressure is constant and fixed at the arbitrary value of 0, the velocity is 0 at all points except for the top boundary which is  $1\text{m/s}$ .

### Determination of the properties at the boundaries of the nodes

The process of computing the variables at the correct position is done at the function calc\_vel, while the functions calc\_derivadesx and calc\_derivadesy determine the value of the required derivatives. The processes that are required are explained at the section Solution using a staggered mesh.

### Determination of the pressure map

To compute the pressure map the following process has to be followed: First  $\mathbf{R}(\mathbf{u})$  has to be computed in the functions calc\_Ru and calc\_Rv. Then the time step has to be determined in the function calc\_inct. Then with this data the predictor velocity ( $\mathbf{u}_P$ ) is determined in the function calc\_up. Finally the coefficients of the Poisson solver are determined in cPoison (constant coefficients, determined at the first time step) and at the Poisson solver function (Poisson) and the iterative process of determining P is started. A Gauss-Seidel algorithm is used.

### Determination of the new velocity field

Having determined the predictor velocity and the pressure distribution it is now possible to compute the velocity at the next time step. This is done in the function Nova\_vel. Also it is important to determine if the program has reached the stationary state. To do this the function Transitori compares the velocity field with the velocity field at the previous time step. If the largest change in the velocity of a node has changed less than a 0.01% the solution has already been reached and the code exits the calculation process. Otherwise the next time step shall be computed.



## Auxiliary plotting functions

When writing a code such as this one it is very important to be able to visualise the results. For this reason various plotting styles have been used. The goal of the auxiliary functions is to write a .dat file that can be interpreted by the software gnuplot and used to plot the results.

Since the problem solution is a vector field the first plotting style consists in plotting the velocity field with the help of vectors. Also to add a more visual effect the colour of the vector depends on the magnitude of the vector modulus. Therefore, this plotting style combines the vector plot and the colour maps used to plot scalar fields in the previous sections. Also to better understand the evolution of the velocity field an animation representing it has been written saving the values of the velocity field every 0.1 seconds and plotting them with gnuplot, it is available online at:

<https://www.youtube.com/watch?v=voW78bEEM3k&feature=youtu.be>.

The colour of each arrow has to be computed in a rgb format, this is done in the function hsv2rgb, which has been obtained from the web.

An animation of the contour plot is also available online at:

<https://www.youtube.com/watch?v=Bk4jzKzOQmU>

To obtain the streamlines of the problem the matlab function streamslice has been used, since it offers a much more clear solution when it is plotted. For more details please see the corresponding section in the Annexes.

## General program scheme

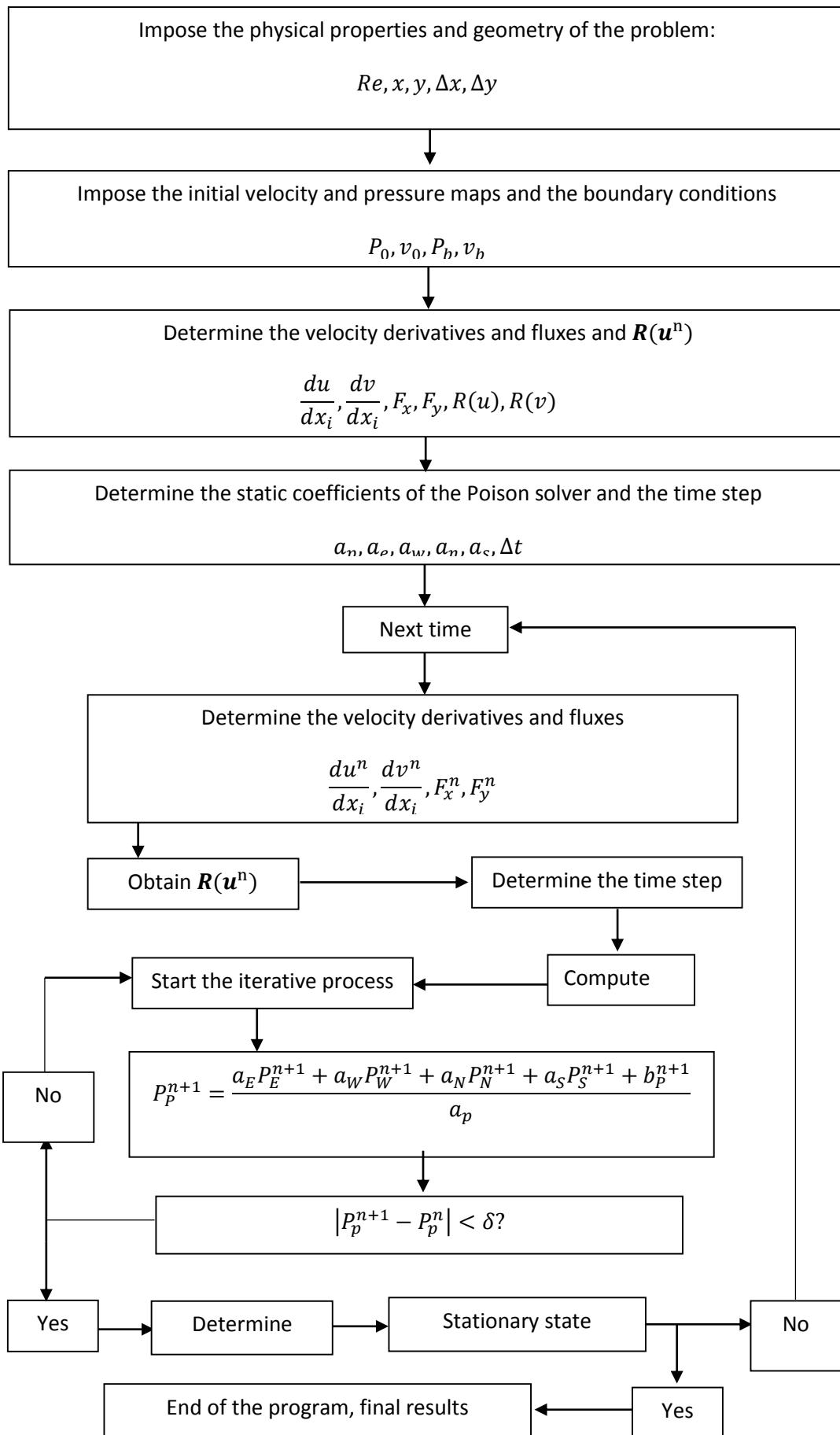
The program computes the functions in the aforementioned order. The only iterative process is produced in the Poisson solver, since the velocity field is determined using an explicit scheme.

The system that has to be solved is very similar to the previous temperature determining processes, but this time the property that is determined is the pressure:

$$a_P P_P^{n+1} = a_E P_E^{n+1} + a_W P_W^{n+1} + a_N P_N^{n+1} + a_S P_S^{n+1} + b_P^{n+1}$$

Then after computing the velocity map the program has to determine if another time step is required and repeat the process.

The program also offers the possibility of solving the driven cavity problem for a number of Reynolds values. These values do not exceed the value of 10000 to ensure that the solution is not affected by turbulent effects.



## Results

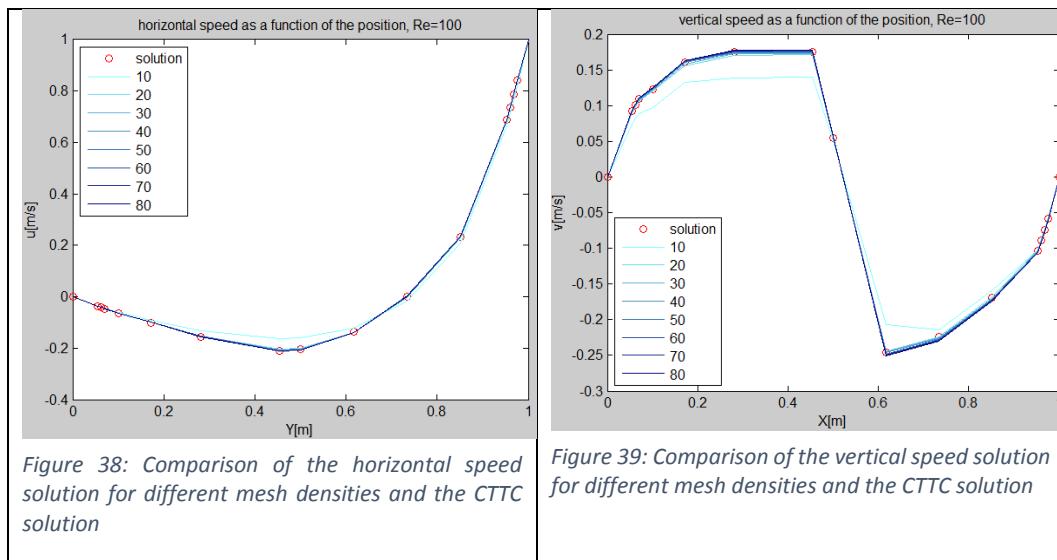
To be able to determine if the results are correct, the main tool that can be used is the numerical solution that the CTTC has provided. This solution contains the value of the horizontal and vertical velocity at some specified points. They can be found in the corresponding annex (Results).

To compare the results with the given table it is important to determine the velocity at the given positions. For this reason the code has the coordinates of the solution nodes implemented and determines by interpolation of the two closest nodes the value of the velocity at such nodes. Then, a Matlab code has been implemented to read the values of the correct solution table and the values of the computed solution and then compare them, this code can be found in the corresponding annex. It is expected that higher mesh densities will offer better solutions.

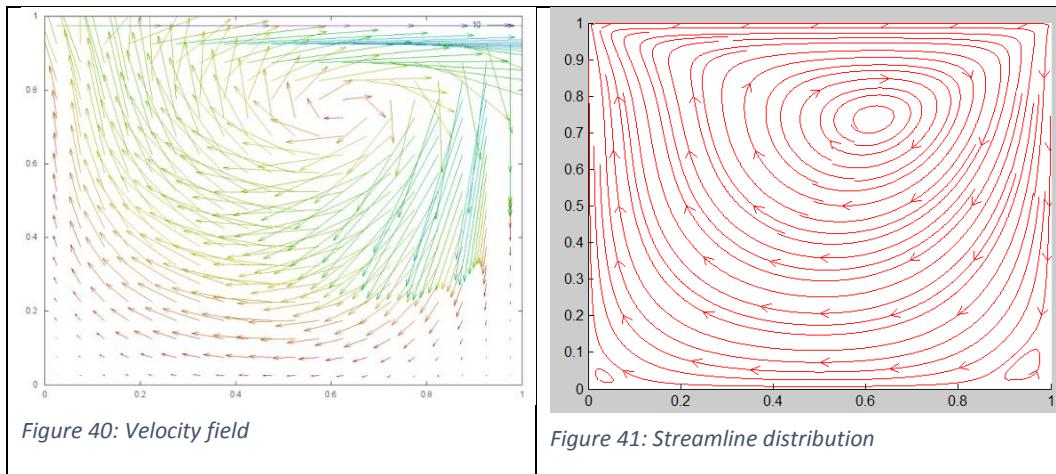
In this section the obtained results are compared with the CTTC solution. This is done for a range of densified meshes that cover from 10x10 nodes to 80x80 nodes. For Reynolds 10000, however, some meshes do not converge with the defined convergence criterion, this is not an error, since for this Reynolds values the solution starts to be turbulent. Therefore for low density meshes it is possible that the solution cannot be obtained with the convergence criterion. For all the other cases the results have been correctly obtained with minimum errors.

### Reynolds 100

As expected, higher mesh densities offer a better solution. In this case meshes ranging from 10x10 to 80x80 nodes for structured meshes have been studied. First of all the solution is compared to the solution provided by the CTTC:



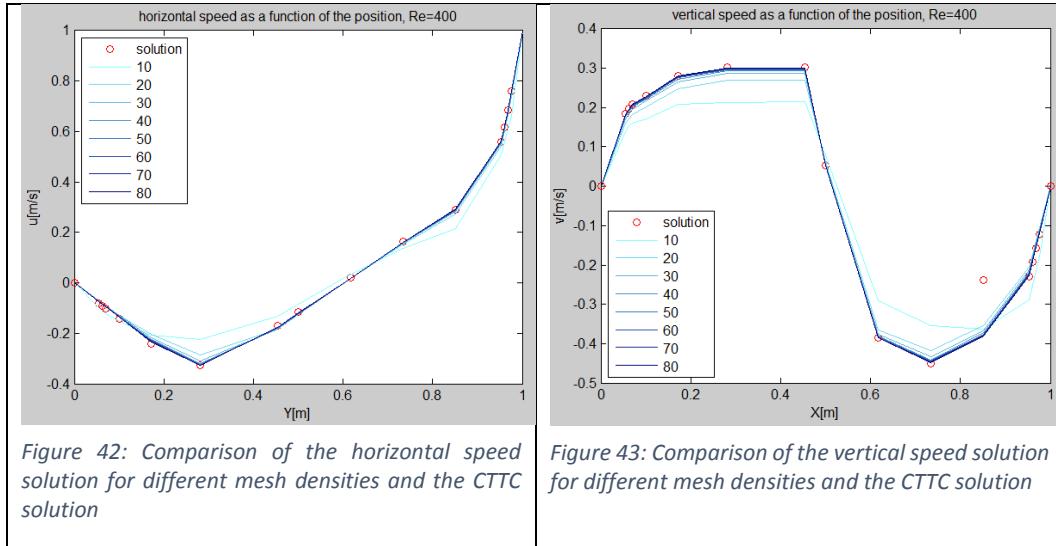
As expected the solution for low number of nodes (in light blue) is very inexact, while with 80 nodes the solution coincides very well with the correct solution. This is true for both the vertical and horizontal velocities, being the horizontal velocity the most precise solution while the vertical solution offers very small errors (<4%) that have been accounted to imprecisions of the numerical technique. This is due to slight imprecisions of the positions of the nodes due to rounding of the program. Still, the solution is very accurate and only has small errors in the nodes situated in the region  $x \in (0.6, 0.9)$ .



By studying the streamline distribution it can be seen that additionally to the central vortex, two other vortex appear in the bottom corners. It is interesting to note that the right bottom corner vortex is much bigger than the left bottom vortex, an evidence that is related to the position of the central vortex, which is clearly closer to the left top corner than the other three corners.

### Reynolds 400

As expected, higher mesh densities offer a better solution. In this case meshes ranging for 10x10 to 80x80 nodes for structured meshes have been studied. First of all the solution is compared to the solution provided by the CTTC:



As expected the solution for low number of nodes (in light blue) is very inexact, while with 80 nodes the solution coincides very well with the correct solution. This is true for both the vertical and horizontal velocities, in this case it is very easy to observe that the node situated approximately at  $X = 0.85$  is wrong for the solution. This is due to a typing error of the original CTTC article.

By studying the streamline distribution it can be seen that additionally to the central vortex various vortex appear by the 2 bottom corners. It is interesting to note that in the right bottom corner the vortex is much more intense than in the left bottom corner. Also it is interesting to note that the central vortex has moved towards the centre of the cavity with respect to the previous case.

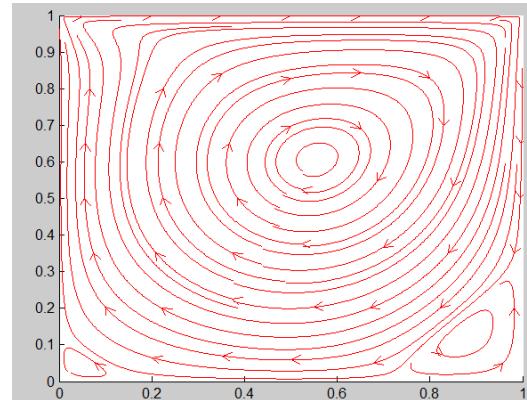


Figure 44: Streamline distribution

### Reynolds 1000

As expected, higher mesh densities offer a better solution. In this case meshes ranging for 10x10 to 80x80 nodes for structured meshes have been studied. First of all the solution is compared to the solution provided by the CTTC:

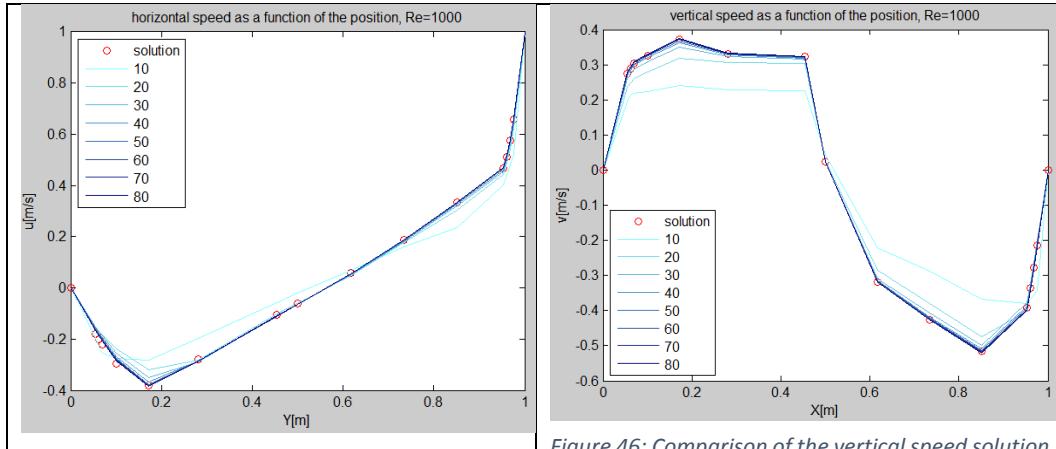


Figure 45: Comparison of the horizontal speed solution for different mesh densities and the CTTC solution

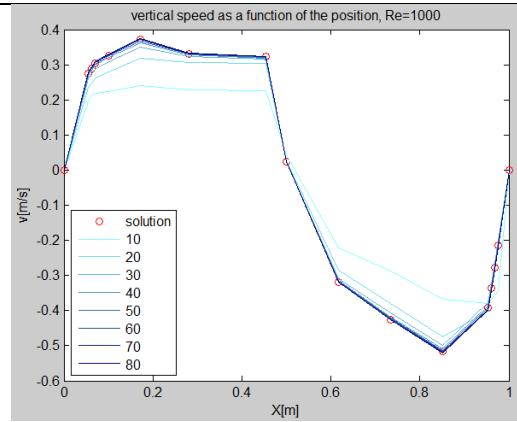


Figure 46: Comparison of the vertical speed solution for different mesh densities and the CTTC solution

As expected the solution for low number of nodes (in light blue) is very inexact, while with 80 nodes the solution coincides very well with the correct solution. This is true for both the vertical and horizontal velocities, being the vertical velocity the most precise solution while the vertical solution offers very small errors (<4%) that have been accounted to imprecisions of the numerical technique. This can be due to the need of a more concentrated mesh for this Reynolds values. Still, the solution is very accurate and only has small errors in the nodes situated in the region  $y \in (0,0.1)$ , and the solution is tending to the CTTC solution.

By studying the streamline distribution it can be seen that additionally to the central vortex various vortex appear by the 2 bottom corners. It is interesting to note that in the right bottom corner the vortex is more intense than in the left bottom corner. Also it is interesting to note that the central vortex has moved towards the centre of the cavity with respect to the previous case. Also the top left corner is beginning to exhibit a singularity.

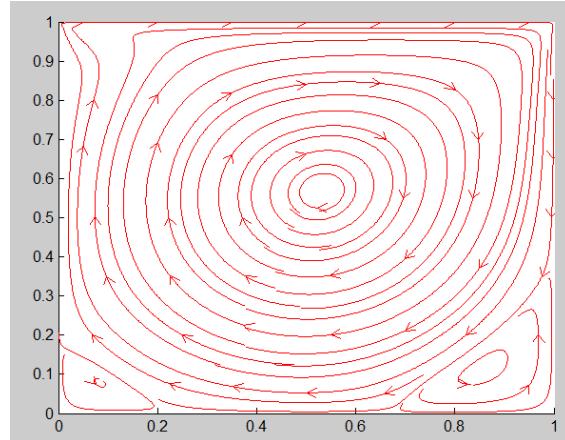


Figure 47: Streamline distribution

## Reynolds 3200

As expected, higher mesh densities offer a better solution. In this case meshes ranging for 10x10 to 80x80 nodes for structured meshes have been studied. First of all the solution is compared to the solution provided by the CTTC:

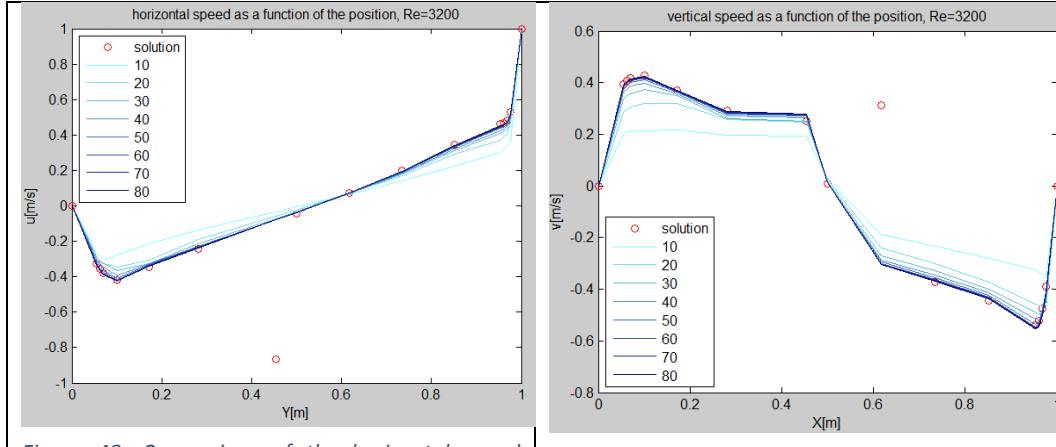


Figure 48: Comparison of the horizontal speed solution for different mesh densities and the CTTC solution

Figure 49: Comparison of the vertical speed solution for different mesh densities and the CTTC solution

As expected the solution for low number of nodes (in light blue) is very inexact, while with 80 nodes the solution coincides very well with the correct solution. This is true for both the vertical and horizontal velocities, in this case it is evident that the solution of the CTTC has two typing mistakes, one for each velocity field. The rest of the nodes coincide nearly perfectly, and a slightly higher mesh density would probably offer a more exact solution, but it is not considered necessary since none of the errors are higher than 4%.

By studying the streamline distribution it can be seen that additionally to the central vortex various vortex appear by the 2 bottom corners but also there is a first hint that a third vortex is appearing in the top left corner. It is interesting to note that the right bottom corner the vortex is the most intense of the three. Also it is interesting to note that the central vortex has nearly reached the geometrical centre of the cavity.

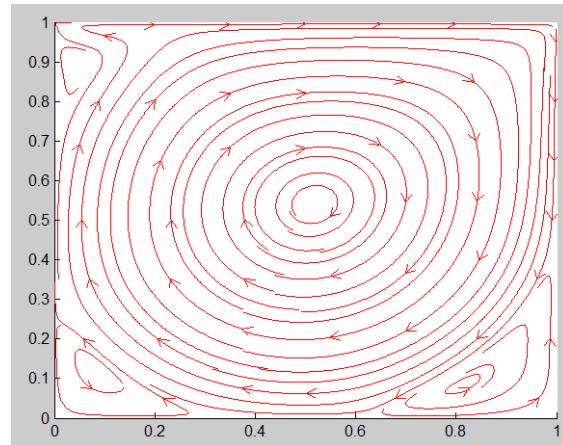
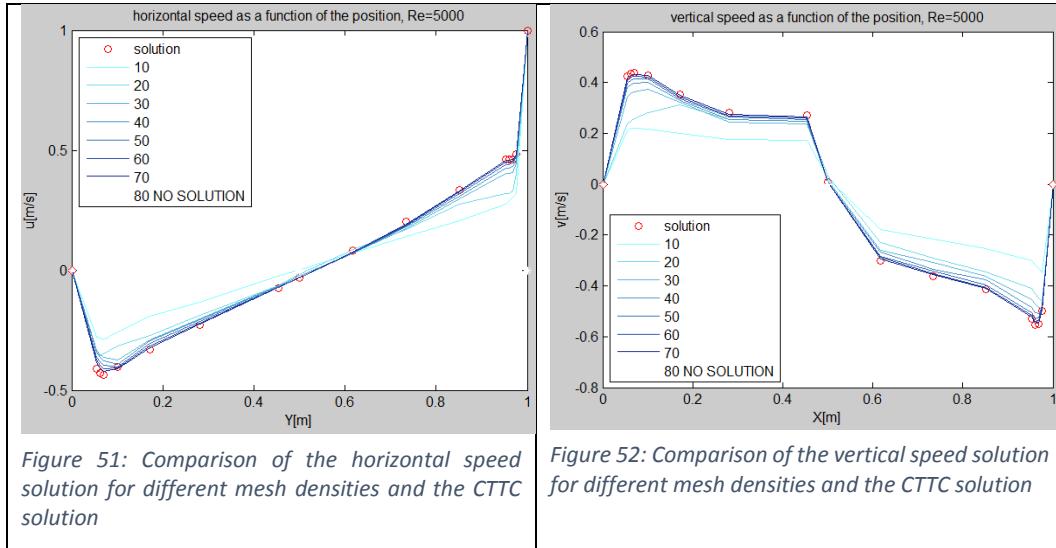


Figure 50: Streamline distribution

### Reynolds 5000

As expected, higher mesh densities offer a better solution. In this case meshes ranging from 10x10 to 80x80 nodes for structured meshes have been studied. First of all the solution is compared to the solution provided by the CTTC:



As expected the solution for low number of nodes (in light blue) is very inexact, while with 80 nodes the solution coincides very well with the correct solution. This is true for both the vertical and horizontal velocities, being the vertical velocity the most precise solution while the vertical solution offers very small errors (<4%) that can be due to the need of a more concentrated mesh for this Reynolds value.

By studying the streamline distribution it can be seen that additionally to the central vortex various vortex appear by the 2 bottom corners. It is interesting to note that in the right bottom corner the vortex is more intense than in the left bottom corner. Also it is interesting to note that the central vortex has moved towards the centre of the cavity with respect to the previous case.

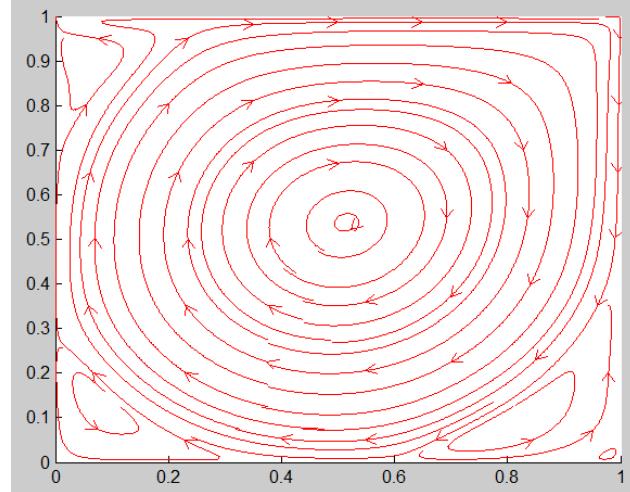
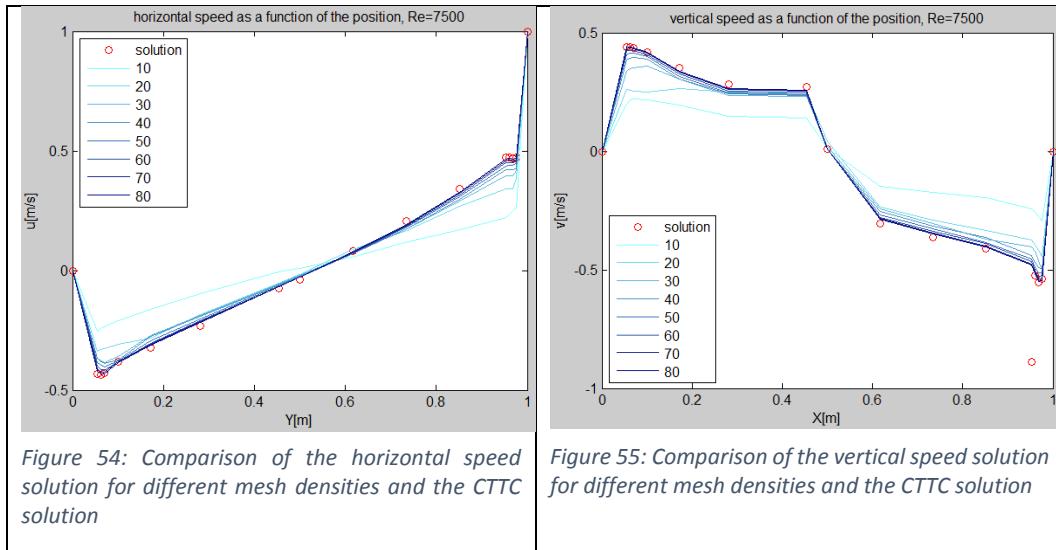


Figure 53: Streamline distribution

Also the top left corner is beginning to exhibit a singularity.

## Reynolds 7500

As expected, higher mesh densities offer a better solution. In this case meshes ranging for 10x10 to 80x80 nodes for structured meshes have been studied. First of all the solution is compared to the solution provided by the CTTC:



As expected the solution for low number of nodes (in light blue) is very inexact, while with 80 nodes the solution coincides much better with the correct solution. This is true for both the vertical and horizontal velocities. However, this solution has slight discrepancies with the CTTC solution. The explanation to this fact is that higher Reynolds values imply a higher turbulent component and therefore denser meshes are required to correctly simulate the problem. If more precision is required a denser mesh can be applied. Still the solution tends to the reference solution and for the 80x80 mesh the highest error is of the order of 6%, which is very good.

By studying the streamline distribution it can be seen that additionally to the central vortex various vortex appear by the 3 corners. It is interesting to note that in the left bottom corner 3 small vortex appear. Also it is interesting to note that the central vortex is situated nearly in the geometrical centre of the cavity.

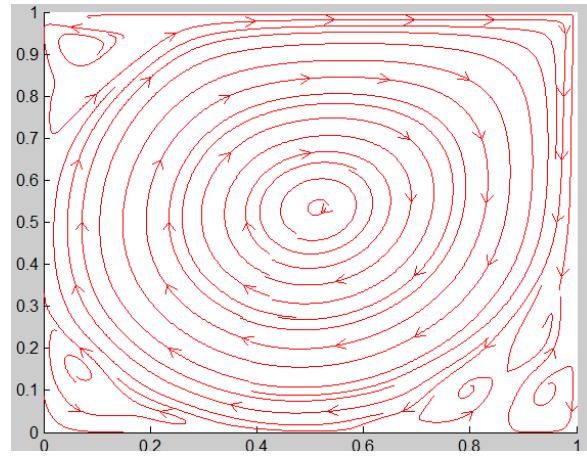


Figure 56: Streamline distribution

## Reynolds 10000

As expected, higher mesh densities offer a better solution. In this case meshes ranging for 10x10 to 80x80 nodes for structured meshes have been studied. First of all the solution is compared to the solution provided by the CTTC:

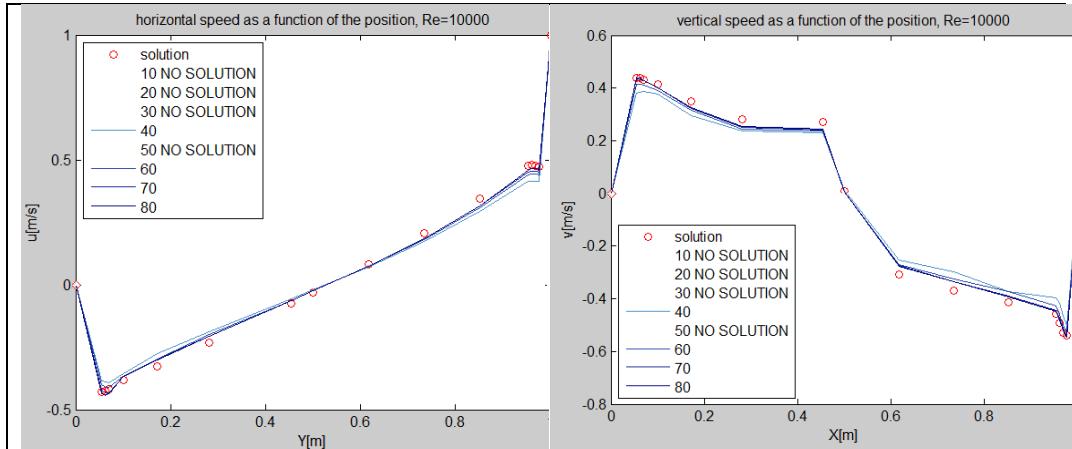
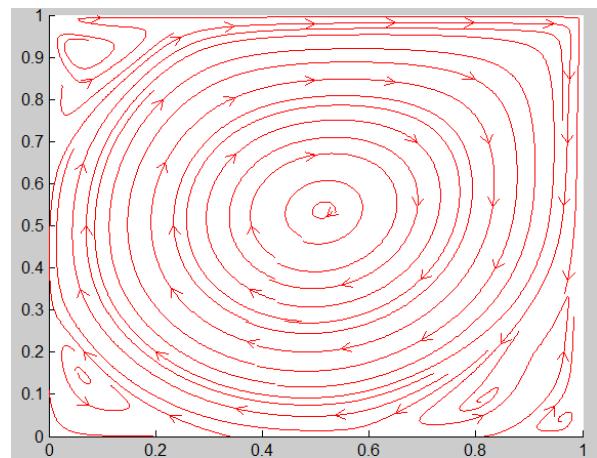


Figure 57: Comparison of the horizontal speed solution for different mesh densities and the CTTC solution

Figure 58: Comparison of the vertical speed solution for different mesh densities and the CTTC solution

As in the previous cases the solution tends to the CTTC solution, this case is probably the least exact solution, but this is due to the turbulent character of this Reynolds value. Also, for this case it is very relevant that some meshes do not converge. This is also explained by the turbulent character of the solution, since low mesh densities cannot account for the turbulent effects: between two time steps the speed derivative will never reach the convergence condition ( $10^{-7}$ ).

By studying the streamline distribution it can be seen that additionally to the central vortex various vortex appear by the 3 corners. Also it is interesting to note that the central vortex is situated nearly in the geometrical centre of the cavity.



## Code verifications

This code implies a high level of complexity. For this reason it is very important to be able to assure that the solution is correct independently of the numerical values obtained: not only must the code converge, but it must also fulfil all the constraints and conditions. In this problem there is a very important constraint around which the whole theoretical is built: the incompressibility of the flux. If this condition is not fulfilled, then the solution will lack any relationship with the physical problem. For this reason it is very important to



ensure that this condition is fulfilled with a high precision (in numerical methods, the 0 value is unobtainable). In this case the incompressibility of the flux can be checked by computing the net flux of each control volume. If the net flux is lower than the admissible error the incompressibility constraint has been achieved, otherwise the code is not computing the solution correctly.

The incompressibility constraint is in fact the condition that establishes that mass cannot be created nor destroyed in incompressible fluids. Since the fluid cannot be compressed its specific volume cannot vary. Therefore if a control volume does not vary in shape or size the amount of fluid that exists in that control volume cannot change. Then if there is a flux through the boundaries of the control volume the net sum of these fluxes must be equal to 0.

The incompressibility constraint can be easily checked: the absolute value of the net flux of each control volume can be studied, in this case, since the precision of the program is  $10^{-7}$  it is expected that the error should be of this order. However, since the flux is equal to the pressure gradient divided by the surface of the control volume, in some cases the incompressibility constraint can reach an error of  $10^{-5}$  due to the errors being divided by the dimensions of the control volume which are of the order of  $\Delta x \sim \Delta y \sim 10^{-1} - 10^{-2}$ .

In the code the accuracy of the incompressibility constraint is stored in the variable eszero[1] and is offered to the user when the solution of the problem is obtained and also at every time step. Should the value be higher than the commented values then the solution cannot be correct.



# Burgers problem

## Introduction

### About DNS

In the previous exercise the Navier-Stokes equations have been approached with a Direct Numerical Simulation. This means that the problem has been solved directly, at each control volume the Navier-Stokes equations are fulfilled. In order for this solution to match reality the meshes that have to be used are very dense. This means that the computational cost of this method is very high. To avoid this problem the Navier-Stokes equations have to be studied more thoroughly. To do so the Navier-Stokes problem is approached from a different perspective: turbulence.

$$\frac{d\mathbf{u}}{dt} + (\mathbf{u} \cdot \nabla) \mathbf{u} = \frac{1}{Re} \Delta \mathbf{u} - \nabla p$$

$$\nabla \cdot \mathbf{u} = 0$$

But which is exactly the cost of DNS? According to Kolmogorov K41<sup>[3]</sup> the smallest time/space scales that have to be solved to obtain a correct solution scale with the Reynolds number:

$$dt \sim Re^{-1/2}$$

$$dx \sim Re^{-3/4}$$

Assuming a 3D problem the memory requirements are proportional to  $Re^{9/4}$  and the computational cost is proportional to  $Re^{11/4}$ .

And why does the computational cost grow in such a manner? This is due to the triadic interactions: the convective term is transporting energy from large scales to small scales and vice versa.

### Burgers equation in Fourier space

To understand the process of energy transport the study of the Burgers equation is proposed (which are the 1D Navier-Stokes equations).

$$d_t u + u d_x u = \frac{1}{Re} d_{xx} u + f$$

This is a simplified model that shares many of the aspects of the 3D Navier stokes equations.

This study is better performed in the Fourier space, in this space the Burgers equation reads:

$$d_t \hat{u}_k + \sum_{k=p+q} \hat{u}_p i q \hat{u}_q = -\frac{k^2}{Re} \hat{u}_k + F_k, \quad k = 0, \dots, N$$

Then 4 terms appear: the temporal evolution of the velocity, the convective term, the diffusive term and the forcing term.

In Fourier space there are no physical dimensions, instead the space is divided in modes, which are related to the frequency. Then high value modes are associated to high frequencies, which are directly related to small scales. Note that this equation is written in the  $\mathbb{C}$  space and therefore  $\hat{u}_k(t) \in \mathbb{C}$  which denotes the k-th Fourier coefficient of  $u(x, t)$ :

$$u(x) = \sum_{k=-N}^{k=+N} \hat{u}_k e^{ikx}$$

The analysis of the Burgers equation leads to the following conclusions:

- 1- The convective term is responsible of the triadic interactions, since for a determined scale  $k$  the scales  $p$  and  $q$  appear.
- 2- The source term is responsible of maintaining the motion, otherwise the equation can be satisfied for all  $\hat{u}_k = 0$ .
- 3- The velocity  $u(x, t) \in \mathbb{R}$  and therefore the condition  $\hat{u}_k = \overline{\hat{u}_{-k}}$  must be accomplished ( $\overline{\cdot}$  denotes the complex conjugate).

#### Kinetic energy transport equation

Kinetic energy can be computed by taking the product of  $\hat{u}_k$  with its complex conjugate  $\overline{\hat{u}_k}$ . Then the Burgers equation can be rewritten as a function of the kinetic energy with the help of the following equation:

$$d_t E_k = d_t (\hat{u}_k \overline{\hat{u}_k}) = \hat{u}_k d_t \overline{\hat{u}_k} + \overline{\hat{u}_k} d_t \hat{u}_k$$

If we denote the convective contribution  $C_k(\hat{u}_p, \hat{u}_q) \in \mathbb{C}$ :

$$C_k(\hat{u}_p, \hat{u}_q) = \sum_{k=p+q} \hat{u}_p i q \hat{u}_q$$

And finally the kinetic energy transport equation can be obtained:

$$d_t E_k = -\frac{2k^2}{Re} E_k - \left( \overline{\hat{u}_k} C_k(\hat{u}_p, \hat{u}_q) + \hat{u}_k \overline{C_k(\hat{u}_p, \hat{u}_q)} \right) + \overline{\hat{u}_k} F_k + \overline{F_k} \hat{u}_k$$

Then two more conclusions can be obtained:

- 1- The diffusive term shall always be dissipating energy. This is due to the fact that the term:

$$-\frac{2k^2}{Re} E_k \in \mathbb{R}^-$$

Also it is interesting to note that the energy dissipation is more effective for high frequency nodes.

- 2- The non-linear convective term is transporting energy mainly from large scales to small scales, but also from small scales to large scales. This is directly related to the triadic interactions:

$$-\left(\overline{\hat{u}_k} C_k(\hat{u}_p, \hat{u}_q) + \hat{u}_k \overline{C_k(\hat{u}_p, \hat{u}_q)}\right) \in \mathbb{R}$$

#### DNS problem approach:

The goal of this problem is to solve the Burgers equation with  $Re = 40$ . To do so the initial conditions are  $\hat{u}_k = k^{-1}$ . Since  $k$  has no interactions with other modes  $\hat{u}_0 = 0$  (no mean flow) is assumed. The results that should be reproduced are contained in Figure 60, both for  $N=20$  and for  $N=100$ . The figure also contains LES solution, which shall be explained in the next stage of the problem solution.

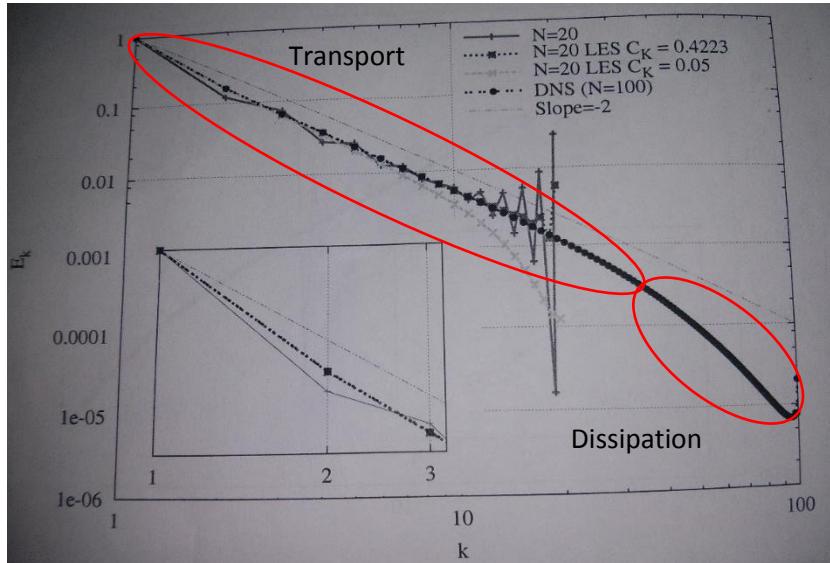


Figure 60: Energy spectrum of the steady state solution of the Burgers equation for  $N=100$  (DNS)

In the energy spectrum it is interesting to identify 2 main regions: the transport region and the dissipation region. In the transport region two factors join: energy is transported more effectively from large scales (low  $k$ ) to small scales (high  $k$ ), and the dissipative term is low. Then energy is mainly transported from these scales to smaller scales. In the dissipation region the effect is different: energy is received from larger scales but the dissipative term is also higher. Therefore energy is mainly dissipated in this region.

## About LES

To reduce the computational cost and the problems that appear when using fewer nodes than those required is to apply a new strategy. This strategy is to try to reproduce the energy cascade assuming a priori properties of the energy spectrum. This technique is named Large Eddy Simulation. The simplest LES model is the Smagorinsky model, but it cannot be applied in the Fourier space. Basically the idea is to add an additional viscosity that shall be a function of  $k$ .

From Figure 60 it is clear that when applying a model with only 20 modes the solution fails to reproduce reality in the larger modes. The proposed model is a spectral eddy-viscosity model proposed by Krachian and improved by Metais and Lesieur<sup>[3]</sup>:

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

With

$$\nu_t^{+\infty} = 0,31 * \frac{5 - m}{m + 1} \sqrt{3 - m} C_k^{-3/2}$$

Where  $m$  is the slope of the energy spectrum, that is  $k^{-m}$ ,  $E_{k_N}$  is the energy at cut-off frequency,  $k_N$ , and  $C_k$  is the Kolmogorov constant.  $\nu_t^*$  is a non-dimensional eddy-viscosity equal to 1 for small values of  $k/k_N$  and with a strong increase for higher  $k$  up to  $\frac{k}{k_N} = 1$ ; it reads:

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

Finally in this case two values of  $C_k$  will be used: 0,05 and 0,4223. And the total viscosity shall be:

$$\nu = \nu_{real} + \nu_t \left( \frac{k}{k_N} \right) = \frac{1}{Re} + \nu_t \left( \frac{k}{k_N} \right)$$

To determine the time step a CFL-like condition must be imposed (taking into account that a fully explicit time-integration scheme is being used).

$$\Delta t < C_1 \frac{Re}{N^2}$$

## LES problem approach

The goal of this problem is to modify the code of the DNS problem and add an additional turbulence term to obtain an energy cascade distribution which is similar to the DNS solution obtained with 100 modes but with only 20 modes. Then both spectres will be compared both in accuracy and computational time.



## Problem solution

### Discretisation of the burger equation

This problem is divided in two parts, but since the second part is only a modification of the first part the vast majority of the program is common to both problem solutions.

The burgers problem reads:

$$d_t u + u d_x u = \frac{1}{Re} d_{xx} u + f$$

In the real space, in Fourier space:

$$d_t \hat{u}_k + \sum_{k=p+q} \hat{u}_p i q \hat{u}_q = -\frac{k^2}{Re} \hat{u}_k + F_k, \quad k = 0, \dots, N$$

Which is much more useful from the programming point of view. It appears that the equation is nearly discretised, since  $\hat{u}_k$  is the k-th mode of the velocity and  $F_k$  is the k-th mode of the forcing term. Then the only term which has to be discretised is the temporal term:

$$d_t \hat{u}_k = \frac{\hat{u}_k^{n+1} - \hat{u}_k^n}{\Delta t}$$

Then a fully implicit scheme the discretised burgers equation reads:

$$\frac{\hat{u}_k^{n+1} - \hat{u}_k^n}{\Delta t} + \sum_{k=p+q} \hat{u}^{n+1}{}_p i q \hat{u}^{n+1}{}_q = -\frac{k^2}{Re} \hat{u}^{n+1}{}_k + F_k, \quad k = 0, \dots, N$$

Which is a system of k equations with k incognita. This system can be iterated and solved in many fashions, in this case the term  $\sum_{k=p+q} \hat{u}^{n+1}{}_p i q \hat{u}^{n+1}{}_q$  will be named "conv" and the term  $\frac{k^2}{Re} \hat{u}^{n+1}{}_k$  will be named "dif". Then, for each mode these terms are computed:

$$\hat{u}_k^{n+1} = \hat{u}_k^n - conv - dif + F_k$$

Taking into account that  $F_k$  is 0 for all modes different than 1, where the velocity does not have to be computed since it is always equal to 1 (if not the solution would be equal to 0 for all modes, at least one mode must have an imposed velocity different to 0). Also mode 0 has its velocity fixed equal to 0 (no mean flow) by the problem definition.

Then the problem shall be:

$$\hat{u}_k^{n+1} = \hat{u}_k^n - conv - dif + F_k, \quad k = 2, \dots, N$$

These operations must be done in the complex space. For this reason the program has to be able to do some simple operations in complex numbers. The functions prod, prodsc,



suma and conj correspond with the product of two complex numbers, the product of a complex number and a scalar value, the sum of two complex numbers and the complex conjugate of a complex number.

To implement the LES solver the numerical viscosity (or eddy viscosity) has to be computed. Then the equation for the DNS solver has to be modified so that:

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

With

$$\nu_t^{+\infty} = 0,31 * \frac{5 - m}{m + 1} \sqrt{3 - m} C_k^{-3/2}$$

And

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

In this case all the variables are known, since  $k_N$  is the cut-off mode, two values of  $C_k$  will be studied,  $m$  is the slope of the linear region of the Energy cascade, which in this case is approximately 2, and  $E_{k_N}$  is the kinetic energy at cut-off which can be computed at each iteration with the value of the velocity of the cut-off mode ( $E_{k_N} = \text{real part}(u_{k_N} * \bar{u}_{k_N})$ ), the kinetic energy is always real, the complex part will always be equal to 0.

Then with all this data the eddy viscosity can be computed for each value of  $k$  at each iteration. To compute the total viscosity, the real viscosity and the eddy viscosity have to be added:

$$\nu = \nu_{\text{real}} + \nu_t \left( \frac{k}{k_N} \right) = \frac{1}{Re} + \nu_t \left( \frac{k}{k_N} \right)$$

So the DNS burgers equation will then become:

$$\frac{\hat{u}_k^{n+1} - \hat{u}_k^n}{\Delta t} + \sum_{k=p+q} \hat{u}^{n+1}{}_p i q \hat{u}^{n+1}{}_q = -k^2 \hat{u}^{n+1}{}_k \left( \frac{1}{Re} + \nu_t \left( \frac{k}{k_N} \right) \right) + F_k,$$
$$k = 0, \dots, N$$

And can be solved identically to the DNS burgers equation:

$$\hat{u}_k^{n+1} = \hat{u}_k^n - conv - dif$$

For this solvers to function correctly an initial velocity map is required. As the problem reads the initial velocity condition is:



$$u_k^0 = \frac{1}{k}, \quad k = 1, \dots, N, \quad \text{and,} \quad u_0^0 = 0;$$

### Burger equation solver

To implement both solvers (DNS and LES) the function burguer has been designed. It is divided in two stages: in the first stage the viscosity is determined taking into account the solution method (DNS or LES). This value is stored in the matrix valor which has N positions. Then, in the second stage the value of the velocity field is computed, since a fully implicit scheme has been used, it is not necessary to implement an iterative process. Still, this process is not implemented in this function, this function only determines the new velocity field at n+1 with the available velocity field (at n). The iterative process will repeatedly call this function and then determine if the solution has reached the steady state.

### Main program

The main program has three main functions: the first one is to call the function burguer and determine the solution for the next time step. Then the second function is to determine if the steady state solution has been reached. This means that the convective term is equal to the diffusive term (and therefore the time derivative of the velocity is 0).

$$\hat{u}_k^{n+1} - \hat{u}_k^n = -conv - dif, \quad k = 2, \dots, N$$

Taking into account that for nodes 0 and 1 the velocity is imposed, then:

$$\text{if } \frac{du}{dt} = 0 \text{ then } 0 = -conv - dif = 0$$

$$-conv = dif$$

And therefore:

$$conv^2 - dif^2 = 0$$

Note that this process is used to avoid working further in the complex space, the verification of the steady state can be done with any of the previous equations.

The third function is to write a .dat file and save the value of the kinetic energy of the steady state solution for each mode there.

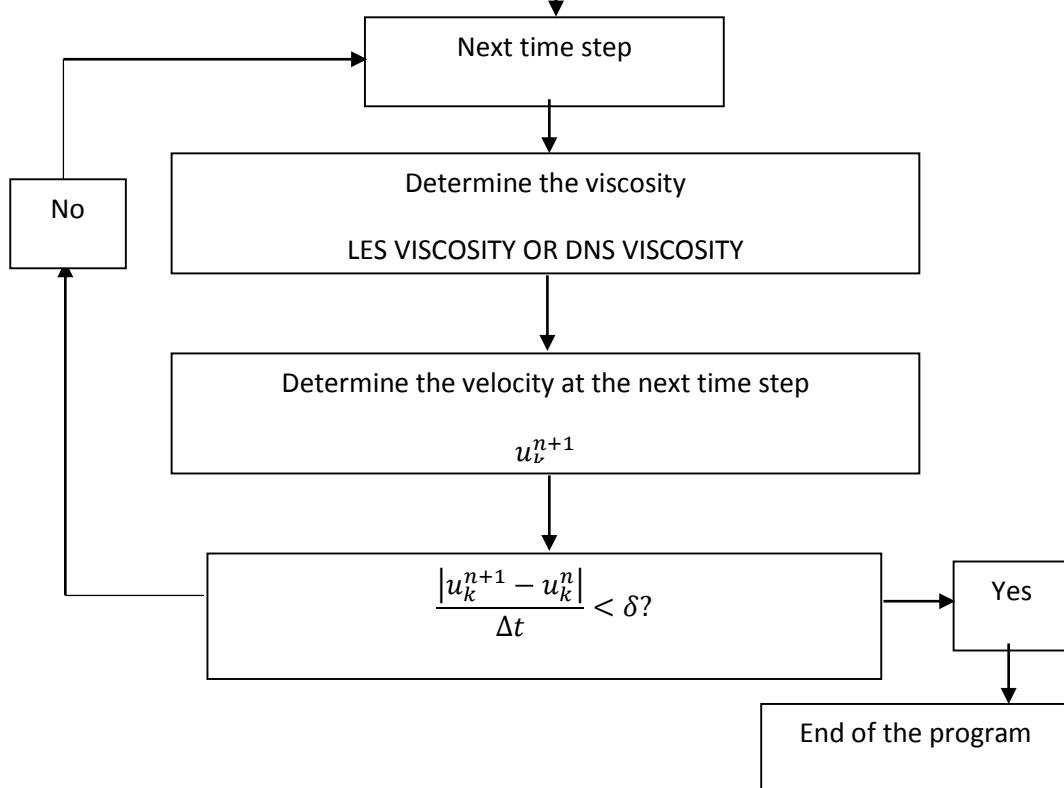
This process has to be repeated for the LES and DNS solutions to elaborate the results that are presented in the results section. The DNS solution has been run for N=20 and N=100 in order to see the consequences of under solving the problem (two few nodes) respect the correct solution. Also the DNS solution has been run with two different values of  $C_k$  to better understand the effect of the additional eddy viscosity.

### General program scheme

The general program scheme is very simple: the main function calls the burguer function in order to determine the velocity at the next time step. Since this velocity is only dependant on the velocity at the previous time step, an iterative process is only required for the time evolution of the velocity field. The program then computes as many time steps as necessary to reach the steady state solution, condition that is evaluated at the main program.

Impose the physical and numerical constants, initial velocity field and determine the time step:

$$Re, N, N_c, C1, C_k, m, u_k^0, \Delta t$$



## Results

The main goal of this program consists in reproducing the energy cascade spectra that can be found in Figure 60.

### DNS results

The first result that will be reproduced is the DNS solution for 20 modes. From Figure 61 it is possible to see that this case is clearly unresolved, the energy cascade is not stable and there are great changes between the solution of one node and the other. This fact

becomes more important with the increasing  $k$ , and for  $k = k_c$  it is clear that the solution is not good enough.

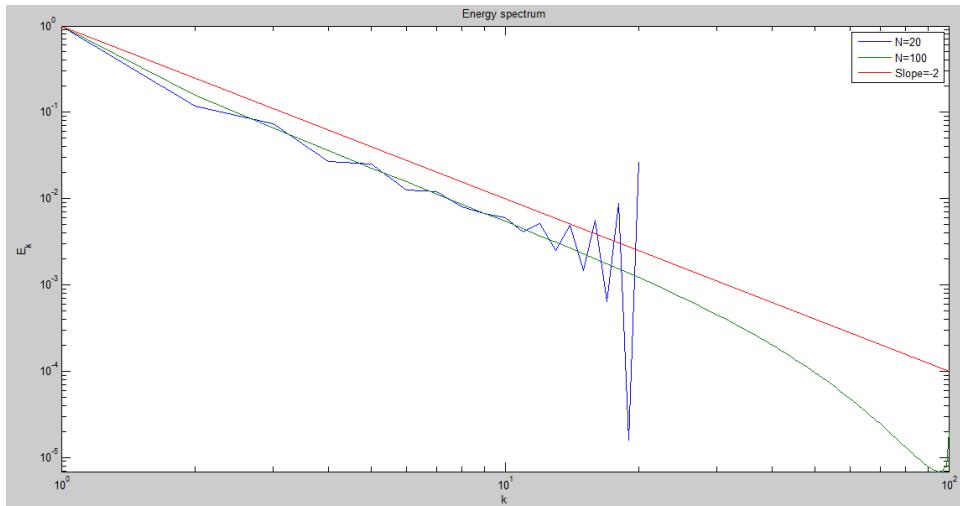


Figure 61: DNS solution for comparison

Secondly the DNS solution for 100 nodes is reproduced. In this case the solution is extremely good, the energy cascade develops in a stable fashion and develops as expected. Two differentiated regions can be found. For low values of  $k$  a linear region with a slope of approximately -2 can be found in which energy is mostly transported into smaller scales, and for high values of  $k$  a non-linear region can be found in which energy is mostly dissipated. The transition appears approximately at  $k \approx 20$  (see Figure 61).

### LES results

Now the main goal is to reproduce the results with a 20 node LES solution that resembles the DNS solution as much as possible. For this reason two values of  $C_k$  have been used. First a value of 0.05 has been used. From the DNS solution for  $N=100$ , which is the exact solution, we expect that the first 20 modes should follow a linear tendency of slope -2. Then from Figure 62 the results coincide very well until mode 12-13, but then the eddy viscosity term becomes too large and the kinetic energy falls too soon, losing the linear trend that was expected.

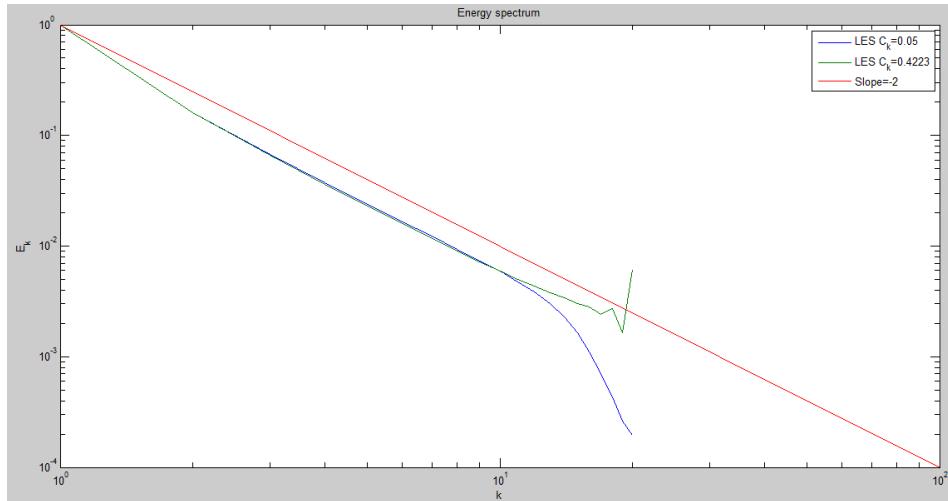


Figure 62: LES solution with  $N=20$  comparison

To reduce the effect of eddy viscosity the value of  $C_k$  has been raised to 0.4223. Then the LES solution with 20 modes should fall less abruptly for large values of  $k$ . From Figure 62 it is clear that the eddy viscosity term is acting less and the tendency to fall prematurely has been reverted. Instead, for this value of  $C_k$  the solution would still have a slight tendency to diverge, but excluding  $k=20$  all the other modes reproduce a very well the DNS solution.

### Comparison of the numerical methods

From Figure 63 it seems clear that the DNS solution with 100 modes is the best solution since it is exact. This, though, comes with a high computational price. It also seems clear that the DNS solution cannot be applied with 20 modes since it is clearly diverging. LES solutions offer the possibility to numerically increase the viscosity and therefore reduce the divergence of the DNS solution. Still, the value of  $C_k$  of LES solution has to be adjusted, for  $C_k = 0.05$  it is clear that the Eddy viscosity is too high and the solution dissipates more energy than it should be. On the other hand a value of  $C_k$  of 0.4223 is slightly too high and in this case the eddy viscosity is slightly lower than it should be and therefore a slight divergence of the solution appears for high values of  $k$ .

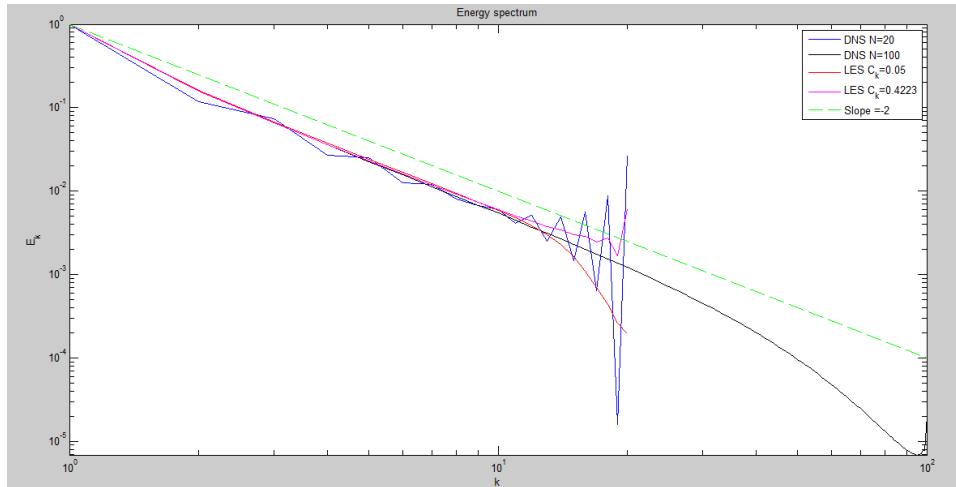


Figure 63: Comparison of the DNS and LES solutions

### Alternative values of $C_k$

In the previous section two values of  $C_k$  were analysed and it was determined that the value of 0.05 had the effect of over estimating the eddy viscosity and that the value of 0.4223 had the effect of slightly underestimating it. In this section an intermediate value of 0.18 is proposed as an alternative, since it resembles more exactly the DNS solution.

As in the previous section the slope is approximately -2, and the behaviour is similar to the case of  $C_k = 0.4223$  (a slight divergence appears for the last node).

If the comparison with the other methods is repeated:

From Figure 65 it seems clear that the intermediate value of 0.18 offers a better approximation to the DNS solution than the previous values.

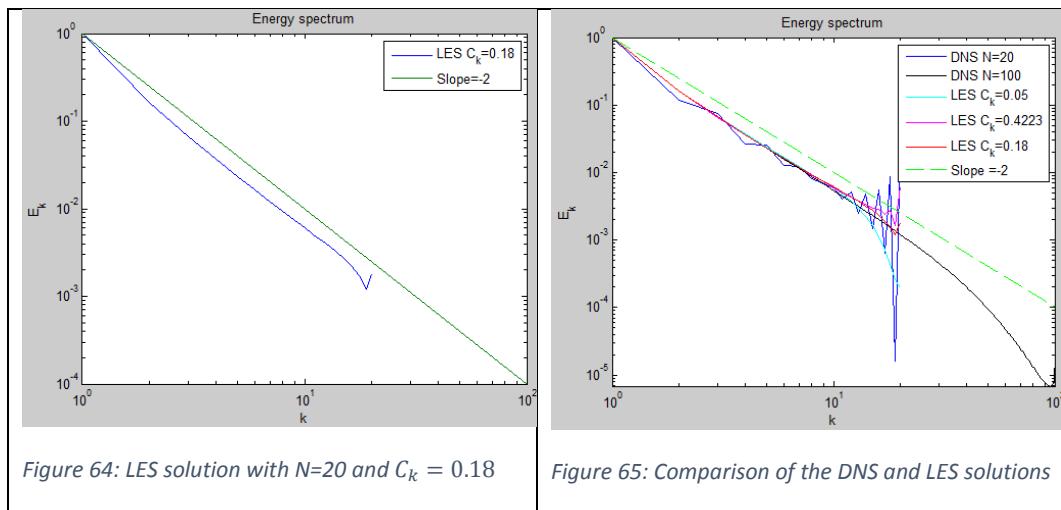


Figure 64: LES solution with  $N=20$  and  $C_k = 0.18$

Figure 65: Comparison of the DNS and LES solutions

# Flux around a square cylinder

## Introduction

As the final problem to solve the flux around a square cylinder will be studied. The main goal is to modify the driven cavity code so that it includes a square cylinder and two walls, with an inlet and an outlet. With this flux many important aerodynamic variables can be studied, such as the drag and lift forces. This kind of simulations offer an alternative to wind tunnel experiments. Figure 66 represents the geometry of the proposed problem<sup>[10]</sup>:

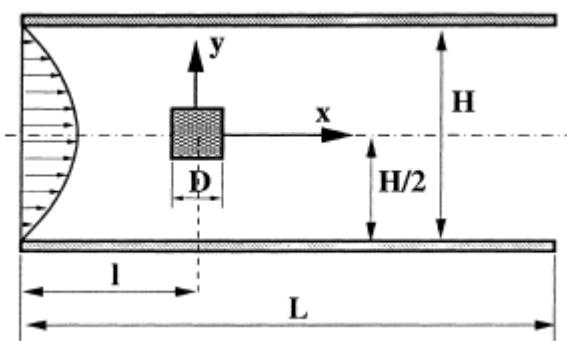


Figure 66: Square cylinder problem geometry

The equations that have to be solved, as in the driven cavity problem, are the Navier Stokes equations.

The boundary conditions include two walls (upper and lower wall) an inlet with a parabolic velocity profile and an outlet where all the physical properties are not altered.

$$\text{inlet: } u(y) = \frac{u_0}{H^2} * (y - H) * y \text{ [m/s]}$$

Where  $u_0$  is the velocity at the centre of the parabolic profile and  $H$  the height of the wind tunnel (see Figure 66).

The outlet does not have a velocity profile associated, and will depend on the velocity of the previous nodes. Then the boundary condition of the outlet is that the derivate of all the velocity component is equal to zero.

$$\frac{d\mathbf{u}}{dx} = 0, \quad \text{outlet}$$

This outlet condition has proved to be more challenging than expected. The solution has come by imposing the velocity of the outlet node equal to the velocity of the previous node on the previous iteration:



$$\mathbf{u}_{outlet}^{n+1} = \mathbf{u}_{previous}^n$$

To include the obstacle a source term has to be included in all the positions where there is no flux (because there is an obstacle) that ensures that the predictor velocity is 0. Then:

$$\mathbf{u}_p^o = \mathbf{u}^n + \Delta t \left( \frac{3}{2} \mathbf{R}(\mathbf{u}^n) - \frac{1}{2} \mathbf{R}(\mathbf{u}^{n-1}) \right) + \mathbf{F}^o = 0$$

This can also be imposed in a similar way that walls are imposed: by establishing that  $\mathbf{u}_p = 0$  and that the flux does not depend on the pressure of the wall (the wall will match the pressure of the flux in each contact surface which shall affect the Poisson solver coefficients).

To compute the aerodynamic forces the definition of such forces is required:

- Lift: Aerodynamic force that is perpendicular to the incident flux. In this case the incident flux is horizontal and therefore the lift is the vertical force (y axis).
- Drag: Aerodynamic force that is parallel to the incident flux. In this case the drag will be the horizontal component of the aerodynamic forces (x axis).

Then the aerodynamic forces have to be computed as a function of the pressure distribution. Recall that:

$$\mathbf{F}_a = -P * S * \mathbf{n}$$

In this case the normal vectors are very easy to compute:

$$\mathbf{n}_{left\ wall} = (-1,0), \quad \mathbf{n}_{right\ wall} = (1,0)$$

$$\mathbf{n}_{bottom\ wall} = (0,-1), \quad \mathbf{n}_{top\ wall} = (0,1)$$

Finally the aerodynamic forces can be computed:

$$L = P_{bottom} * S_{bottom} - P_{top} * S_{top}$$

$$D = P_{left} * S_{left} - P_{right} * S_{right}$$

Which for the discretised domain (only the nodes belonging to the boundary of the cylinder):

$$L = \left( \sum_{i_0}^{i_f} P_i * S_i \right)_{bottom} - \left( \sum_{i_0}^{i_f} P_i * S_i \right)_{top},$$
$$D = \left( \sum_{j_0}^{j_f} P_j * S_j \right)_{left} - \left( \sum_{j_0}^{j_f} P_j * S_j \right)_{right}$$

## Numerical solution

The numerical solution of this case uses the same code than the driven cavity solver, with some slight changes:

First of all the boundary conditions have changed, this has a very small impact on the solver, the only modification is to change the predictor velocity at the boundary. However, the boundary condition of the outlet has an undesired effect: since the balance of the control volumes has the precision of the solver, a slight discordance is added after each iteration. This error is of the order of the error of the precision of the solver, and after a number of time steps it can grow and affect the solution, since the mass of the domain will not be conserved. To avoid this, at the beginning of each iteration the total mass flow of what should be the outlet velocity is computed. Then, the difference between the outlet and the inlet flows is determined and divided between the outlet nodes so that the total mass of the domain is perfectly conserved.

Secondly, since the cylinder is not a fluid, the Poisson coefficients of the cylinder nodes are all reduced to 0 except for  $a_p$ . This way, the pressure at the cylinder interior is not computed. By doing this  $\mathbf{u}_p$  is equalled to 0 in all the velocity components, and no flux can go through the cylinder. The Poisson coefficients of the flux that is directly next to the cylinder are also modified, so that the pressure in such nodes does not depend on the pressure of the cylinder wall. This is equivalent to establishing that no flux from this nodes can go through the node boundary that is shared with the cylinder.

Finally the most important modification to the code is the design of a specific mesh that offers a good mesh density where it is needed with the minimal computational cost.

### Design of the mesh

The domain is a rectangle of 10x1,6m. To design the mesh the domain has been divided in 3 vertical divisions and 4 horizontal divisions.

#### Vertical mesh

3 vertical divisions separate the domain in 3 sections.

The first section covers from the lower wall to the lower part of the cylinder. The mesh that is used in this section is densified. The function that is used to densify the mesh at the bottom wall and the cylinder proximity is the following hyperbolic tangent function:

$$y_b = \frac{y_{cylinder}}{2} * \left( 1 + \frac{\tanh\left(\gamma\left(\frac{2(i-1)}{N}\right) - 1\right)}{\tanh\gamma} \right)$$

The second section covers the height of the cylinder and uses a regular mesh with a high node density. This is done with the goal of describing the velocity field near the cylinder in the best possible manner. Finally, the third section uses the same mesh distribution as the first section. By doing this a symmetrical mesh is defined.

### Horizontal mesh

4 horizontal divisions separate the domain in 4 sections: The first section covers from the inlet to the left wall of the cylinder. It is defined by a densified mesh that is denser near the cylinder and the inlet. It follows the same formula than the vertical mesh.

The second section covers the width of the cylinder and uses a regular mesh with a high node density. This is done with the goal of describing the velocity field near the cylinder in the best possible manner.

The third section covers the wake of the cylinder and uses a regular mesh with the same density as in the previous section. The length of this section is 3 times the length of the cylinder. This section has a very high computational cost, but it is necessary to do so to obtain a correct solution of this problem.

The last section covers the part between the end of the wake and the outlet. To reduce the computational cost the number of nodes has been reduced in this section using the same densifying function. The mesh gradually becomes less dense and densifies again near the outlet.

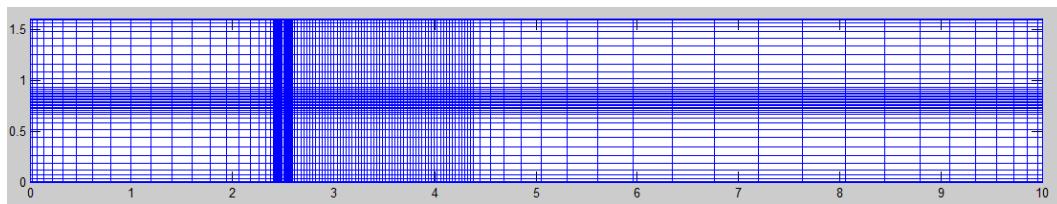


Figure 67: Mesh representation

## Results

In this problem it is very interesting to compare how the flux behaves as the Reynolds number varies. For low Reynolds values the flux is expected to be stable, with laminar behaviour and very clear streamlines that avoid the cylinder and then close symmetrically. As the Reynolds number increases the flux is expected to behave in a less stable way, turbulence will begin to appear and the solution will not reach a stable state. The wake will then be very interesting and the streamlines that go through it will offer an unsteady solution similar to the von Karman vortex sheet. This solution consists in an oscillating unsteady wake that is responsible among many others of the waving of flags or the

vibration of suspended cables that emit a very particular sound. However, the von Karman vortex require a turbulence solver, so the analysis will be limited to lower Reynolds values.

These effects are associated to the ratio between the convective and diffusive terms. For low Reynolds values the diffusive term is dominant over the convective term and therefore the transitions are smooth. With increasing Reynolds the convective term becomes more and more important and the diffusive term loses importance. Therefore high velocity gradients appear and the solution becomes unstable leading to turbulent conditions.

To completely solve this problem with a laminar Poisson solver a mesh density of approximately 100 nodes per cylinder face is required<sup>[10]</sup>. This implies that the computational cost of this problem is extremely high, and therefore not all the cases have been reproduced. However with lower density meshes the process that leads to the formation of the wake can be observed, so, for low Reynolds values the solution will be very similar to the correct solution (10-30). With increasing Reynolds values (reaching  $re = 60$ ), however, the solution will not behave as expected, since the mesh is not dense enough to simulate the von Karman vortices.

In this section, the solution will be given for Reynolds values that try to reproduce the results from the CTTC<sup>[10]</sup>, which are Reynolds 1, 30, 60 and 200, although the last two cases have not been reproduced with an adequate mesh.

### Reynolds 1

As expected for low Reynolds values the solution is laminar and the flux is recovered shortly after the square cylinder perturbation. This can be easily seen in the streamline distribution (Figure 68) which is nearly symmetrical in the x axis. This case offers a good solution because the flux can be considered laminar, and therefore the mesh is dense enough to correctly describe the velocity field.

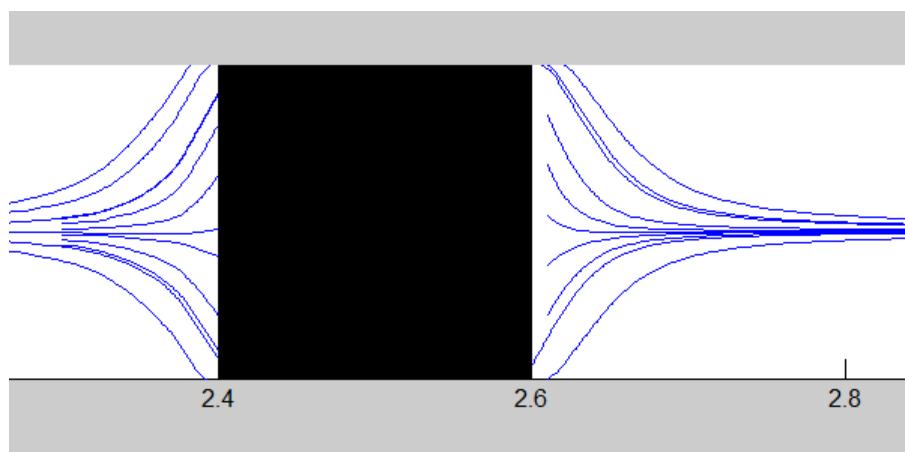


Figure 68: Streamline distribution

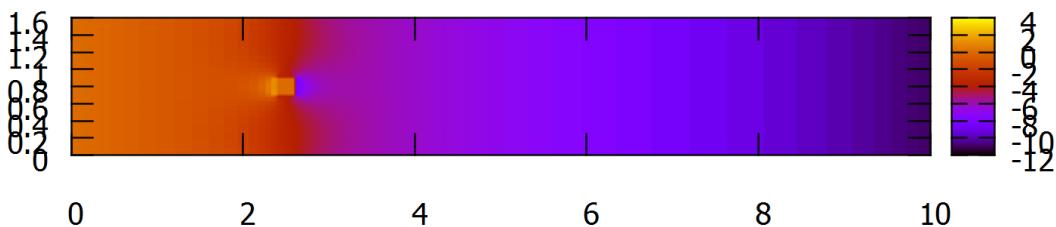


Figure 69: Pressure map

In the pressure map evidence that the boundary layer will detach from the posterior part of the cylinder first appears. However, in the velocity field is very symmetrical, which indicates a very laminar behaviour.

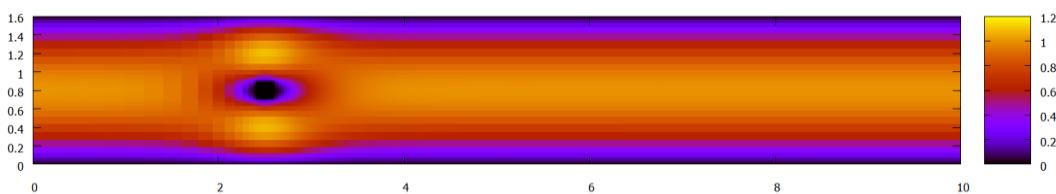


Figure 70: Velocity field

A more dense mesh would probably be advisable, but the computational cost of this Reynolds value grows very fast, and other Reynolds values required more attention.

### Reynolds 20

For this case two meshes have been tried in order to try to reach the solution (40x120 and 150x300). However, the turbulent character of this solution requires even a higher number of nodes or a turbulent solver. Therefore when trying to solve it by the DNS method the solution does not show vortexes for the first mesh, only a small perturbation appears (Figure 71).

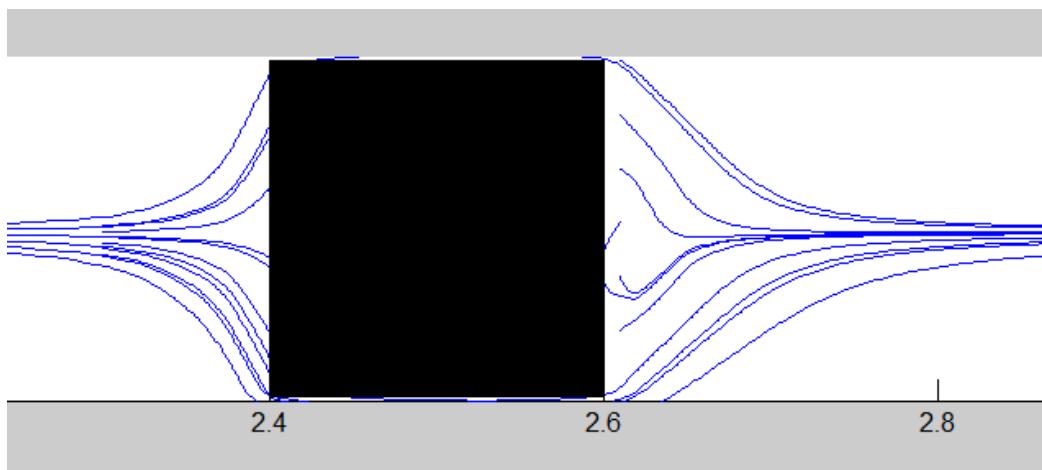
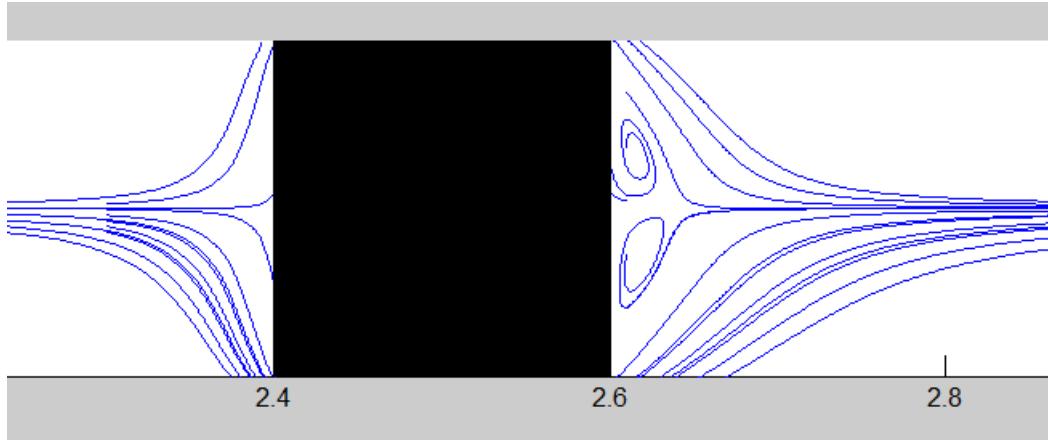
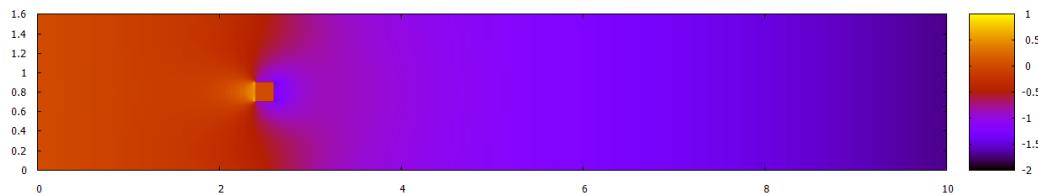


Figure 71: Streamline distribution for the 120x40 mesh



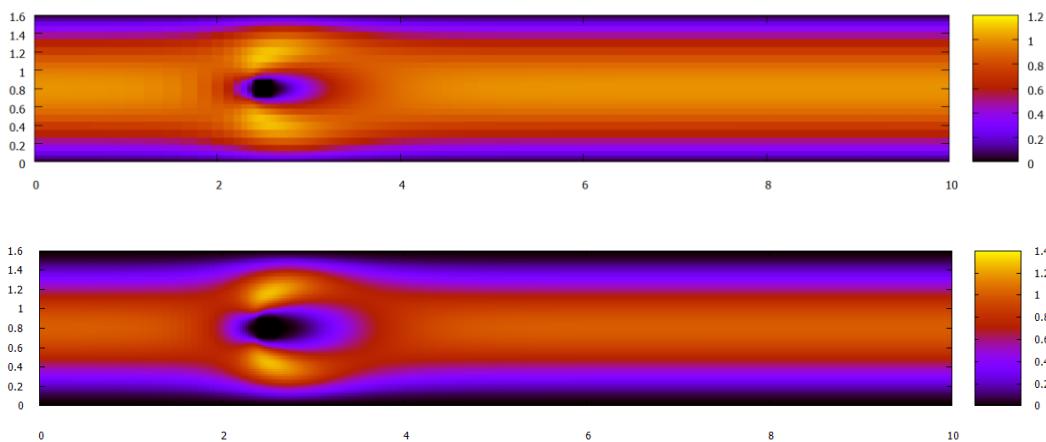
When applying the second mesh, however, two vortexes appear, clearly indicating that the solution is under resolved (see Figure 72). This behaviour is coherent with the theory, which indicates that when a DNS solution is obtained for a mesh that is too coarse, then the solution will tend to underestimate the effects of turbulence. If a thinner mesh is applied, then the vortexes will be even more intense.

*Figure 72: Streamline distribution for the 300x150 mesh*



*Figure 73: Pressure map for the 300x150 mesh*

In the pressure and velocity fields it is clear that the boundary layers are detached from both the sides and the back of the cylinder, this indicates again that this case shall be of a turbulent nature.



*Figure 74: Velocity fields for the 120x40 and 300x150 mesh*

Probably the clearest way to see that the solution is under resolved is by comparing the two velocity fields from Figure 74. Then it is obvious that the wake is much smaller in the low density solution and grows with increasing mesh size.

### Reynolds 40

For this case two meshes have been tried in order to try to reach the solution (40x120 and 150x300). Two more meshes were attempted (200x400 and 300x600), but the computational cost was too high and the schedule did not allow them to be completed on time. The meshes that have been solved are not dense enough and the turbulent character of this solution requires even a higher number of nodes or a turbulent solver to find a more exact solution. Therefore when trying to solve it by the DNS method the solution shows vortexes for the first mesh but they are smaller than expected<sup>[10]</sup> (see Figure 75).

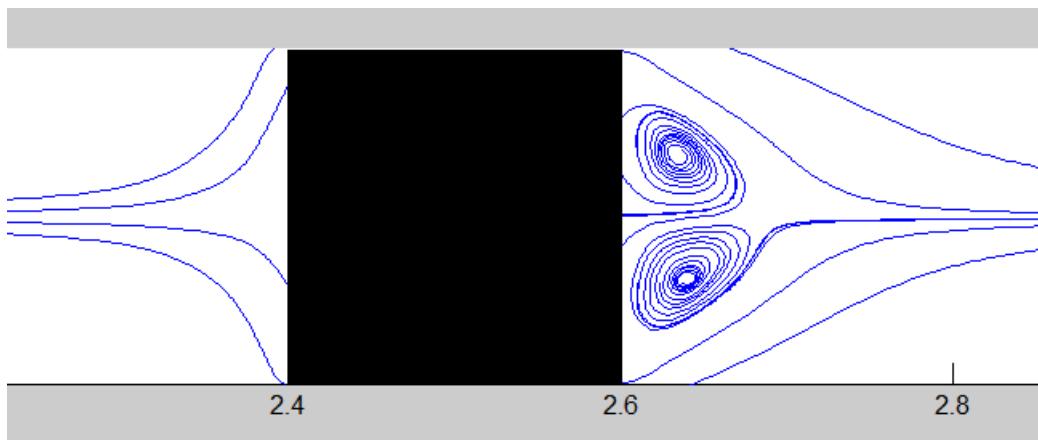
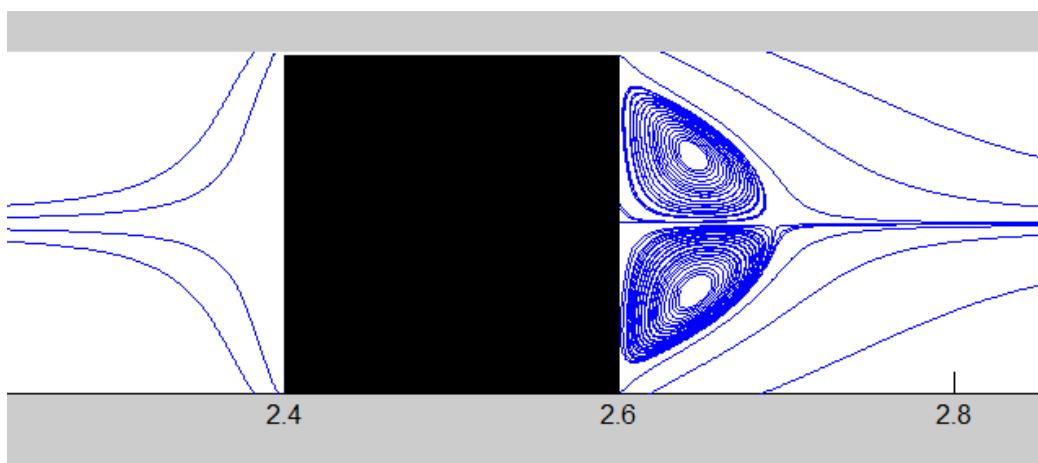


Figure 75: Streamline distribution for the 120x40 mesh



When applying the second mesh, the two vortexes change in shape and become more intense clearly indicating that the solution is under resolved (see Figure 76). This behaviour is coherent with the theory, which indicates that when a DNS solution is

obtained for a mesh that is too coarse, then the solution will tend to underestimate the effects of turbulence. If a thinner mesh is applied, then the vortexes will be even more intense.

Figure 76: Streamline distribution for the 300x150 mesh

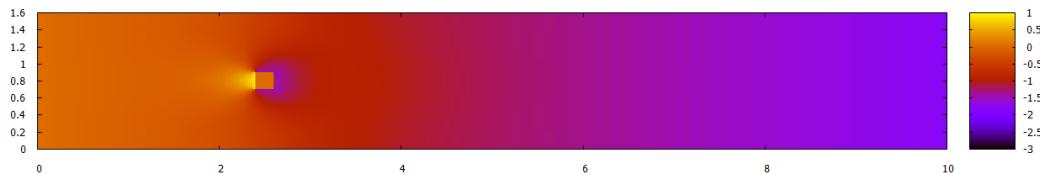


Figure 77: Pressure map for the 300x150 mesh

In the pressure and velocity fields it is clear that the boundary layers are detached from both the sides and the back of the cylinder, this indicates again that this case shall be of a turbulent nature. For this case it is mandatory to run a LES solver or similar to correctly simulate the turbulent flux on the back wall of the cylinder.

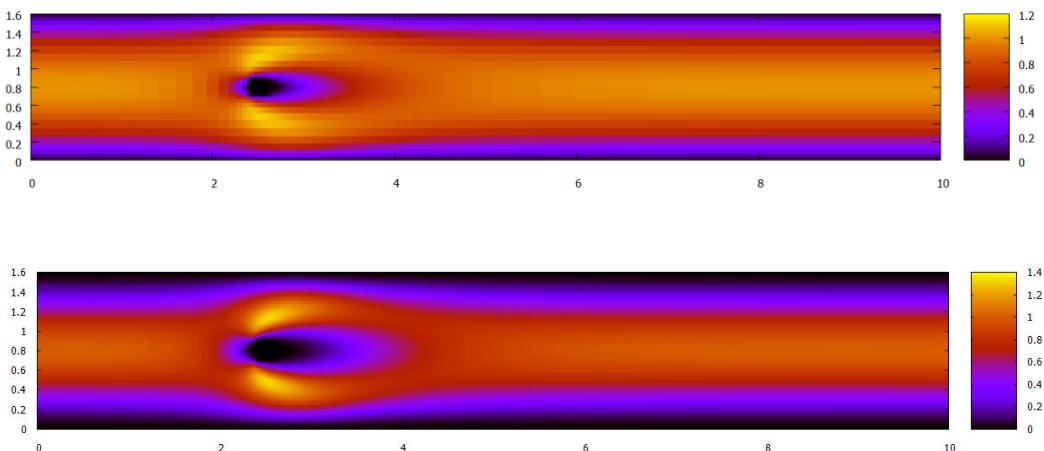


Figure 78: Velocity fields for the 120x40 and 300x150 mesh

Probably the clearest way to see that the solution is under resolved is by comparing the two velocity fields from Figure 78. Then it is obvious that the wake is much smaller in the low density solution and grows with increasing mesh size. With growing Reynolds number the difference between the solutions grows, clearly indicating that the Reynolds number is the key factor that determines the mesh size. This is also coherent with the theory of turbulent flows<sup>[3]</sup>.

## Reynolds 60

For this case two meshes have been tried in order to try to reach the solution (40x120 and 150x300). Two more meshes were attempted (200x400 and 300x600), but the computational cost was too high and the schedule did not allow them to be completed

on time. The meshes that have been solved are not dense enough and the turbulent character of this solution requires even a higher number of nodes or a turbulent solver to find a more exact solution. Therefore when trying to solve it by the DNS method the solution shows vortexes for the first mesh but they are smaller than expected<sup>[10]</sup> (see Figure 79).

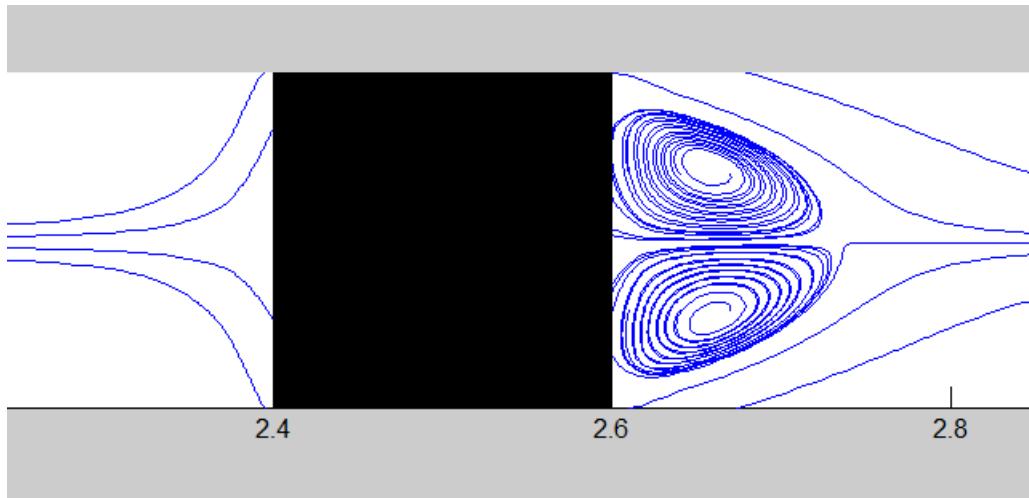


Figure 79: Streamline distribution for the 120x40 mesh

When applying the second mesh, the two vortexes change in shape and become more intense clearly indicating that the solution is under resolved (see Figure 80). This behaviour is coherent with the theory, which indicates that when a DNS solution is obtained for a mesh that is too coarse, then the solution will tend to underestimate the effects of turbulence. If a thinner mesh is applied, then the vortexes will be even more intense.

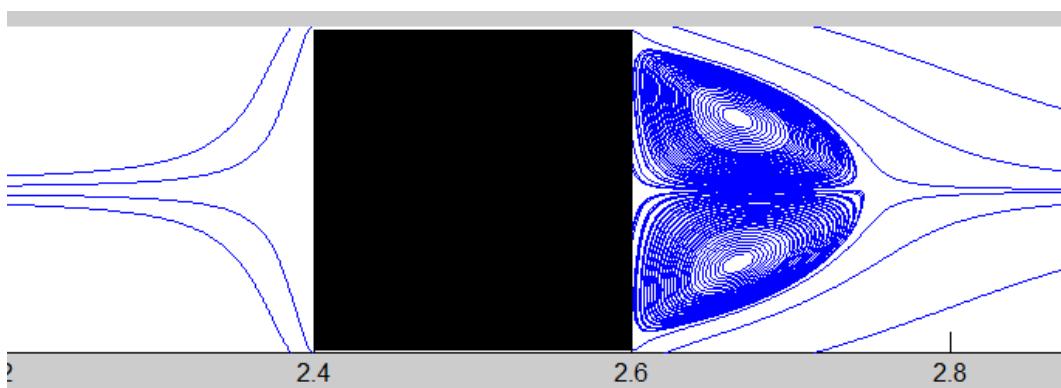


Figure 80: Streamline distribution for the 300x150 mesh

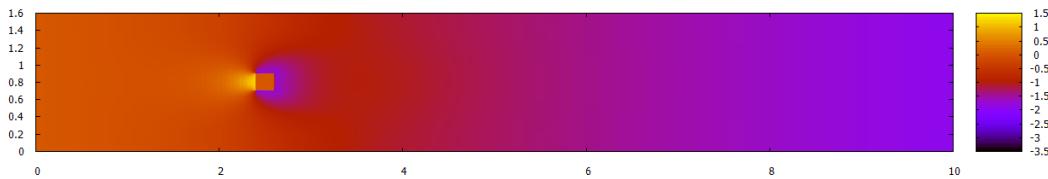


Figure 81: Pressure map for the 300x150 mesh

In the pressure and velocity fields it is clear that the boundary layers are detached from both the sides and the back of the cylinder, this indicates again that this case shall be of a turbulent nature. For this case it is mandatory to run a LES solver or similar to correctly simulate the turbulent flux on the back wall of the cylinder.

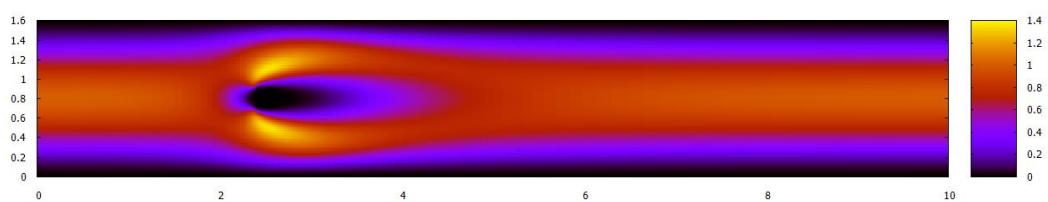
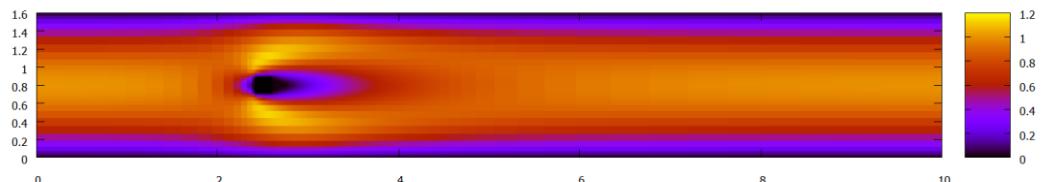


Figure 82: Velocity field for the 300x150 mesh

Probably the clearest way to see that the solution is under resolved is by comparing the two velocity fields from Figure 78. Then it is obvious that the wake is much smaller in the low density solution and grows with increasing mesh size. With growing Reynolds number the difference between the solutions grows, clearly indicating that the Reynolds number is the key factor that determines the mesh size. This is also coherent with the theory of turbulent flows<sup>[3]</sup>.

# B4' Scope and specifications performance:

The scope of the project contains 5 main points:

1. Numerical solution and verification of the conduction equations
2. Numerical solution and verification of the convection equations
3. Numerical solution and verification of the Navier-Stokes equations
4. Numerical solution and verification of turbulence
5. Application of the numerical solution of the Navier-Stokes equations to an aerodynamic case containing an object.

For point number 1 the scope has been fully addressed. The conduction equations have been solved and the solution has been verified with the CTTC solution as well as in code verifications.

For point number 2 the scope has been fully addressed. The convection equations have been solved and the solution has been verified with the CTTC solution as well as in code verifications. The solution has been verified for different  $\rho/\Gamma$  values.

For point number 3 the scope has been fully addressed. The Navier-Stokes equations have been solved and the solution has been verified with the CTTC solution as well as in code verifications. The solution has been verified for different Reynolds number values.

For point number 4 the scope has been addressed for 1D turbulence. The burger's equation has been solved and verified both for DNS methods and LES methods. Unfortunately the lack of more time has not allowed the implementation of 2D simulation of turbulence.

For point number 5 the scope has been addressed for laminar simulation, turbulent simulation cannot be addressed without a 2D turbulence solver since the computational time is too high for DNS simulations. With additional time a 2D turbulent solver for the flux around the square cylinder could have been developed. This can be developed in a further study.

All the codes comply with the specifications defined by the CTTC: they compile, execute without any input and offer solutions that are very similar to the correct results. Also it is very important in this kind of simulations that the properties are conserved. This is especially important in the Navier-Stokes solution, where the most important requirement is that mass is conserved.

# B5 Budget and environmental impact

## Budget

The main task of this project consists in programing a CFD code. Therefore the direct costs shall be high (due to a great amount of engineering work) and the indirect costs shall be low (the main tool is a computer and the required software).

Direct costs	5140€
Indirect costs	963,35€
<b>TOTAL</b>	<b>6103,35€</b>

Table 6: Budget

## Environmental impact

This study uses the computer as the main tool. This means that there is an energetic cost, but the goal is to avoid using a wind tunnel, which has a much higher energetic cost. Also the basic student tools that have been used are paper and pens for the deductions. For this reason the following environmental impact causes have been determined:

- Wind/water tunnel costs
- Energy consumption
- Production of the computer
- Paper and other materials

This project consists in simulating the flux around an object. If the simulations are correct, then a wind tunnel experiment can be avoided for each simulation. The environmental costs of a simulation shall always be lower than a wind/water tunnel experiment, since wind/water tunnels are typically noisy and consume a great amount of energy, much higher than the amount of energy required to run a computer. Also if this project avoids the construction of a new wind/water tunnel because all the experiments can be simulated, then the benefits are much higher, both in the construction and in the maintenance.

For the energy consumption the environmental impact is not very high. Two factors contribute to this fact. On the one hand the energy consumption of the computer that has been used is not very high since it includes components that have been designed in the last years and therefore it is optimised to reduce consumption. On the other hand the computer has worked mostly during night time, when the energy production is much higher than the demand due to non-stoppable energy sources such as nuclear power or renewable technologies.



The production of the components of the computer involves a high impact on the environment. Modern computers include many components that are rare or hard to obtain and require high amounts of energy to be processed. According to the newsletter of United Nations University<sup>[11]</sup>:

“The average 24 kg desktop computer with monitor requires at least 10 times its weight in fossil fuels and chemicals to manufacture, much more materials intensive than an automobile or refrigerator, which only require 1-2 times their weight in fossil fuels. Researchers found that manufacturing one desktop computer and 17-inch CRT monitor uses at least 240 kg of fossil fuels, 22 kg of chemicals and 1,500 kg of water a total of 1.8 tonnes of materials.”

Finally the paper and other materials consumption does not involve a particular increase in the needs of a student day life. For this reason this impact is not very relevant and will be neglected.

# B6 Temporal aspects

## Temporal planning

For this section a proposal of the CTTC professor Xavier Trias will be scheduled. It consists in adapting the square cylinder problem so that it includes two cylinders. Then an analysis of the interaction between the wakes of the two cylinders can be performed.

This proposal will mainly consist in four tasks:

Task 1: To redefine the mesh to adapt it to the second cylinder. Since the mesh is much less dense in the posterior part of the domain, the equations would not be correctly solved if the mesh is not adequate. Therefore it has to be redefined. This task has an expected time expense of three days, spending an average of 4 hours a day.

Task 2: To adapt the Poisson solver so that no flux goes through the second cylinder. The solver must understand that the Navier Stokes equations are not valid in the cylinder domain and therefore that no flux can enter the cylinder. Also the lift and drag equations have to be redefined to take into account the second cylinder. This task has expected time expense of three days, spending an average of 4 hours a day.

Task 3: To simulate the new problem, obtain new results, verify them and plot them. Finding the errors and correcting them can only be done by studying the preliminary results, and plotting is required to interpret such results. This task has expected time expense of five days, spending an average of 4 hours a day.

Task 4: To write a brief memorandum of the work that has been completed, including the three task and an interpretation of the results. This task has expected time expense of 6 hours, which can be done in one day.

Taking as a starting date the 1<sup>st</sup> of June, 2015 the following schedule defines the time distribution that can lead to the completion of the tasks that have been defined in two weeks' time:

Task	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>	5 <sup>th</sup>	6 <sup>th</sup>	7 <sup>th</sup>	8 <sup>th</sup>	9 <sup>th</sup>	10 <sup>th</sup>	11 <sup>th</sup>	12 <sup>th</sup>	13 <sup>th</sup>	14 <sup>th</sup>
1														
2														
3														
4														

Table 7: Schedule for the tasks

# B7 Conclusions and recommendations

## Conclusions

This study is very far away from the current state of the art solvers. There are many ways to improve the code and make it more general, so that it can solve more cases. Still, the amount of knowledge that I have gained is invaluable. In a world that is tending to numerical simulations in many fields, it is important not only to understand the concepts behind numerical simulation, but also to have a working knowledge. After completing the course on numerical methods given at the ETSEIAT I felt that although I had a theoretical background I did not have the tools to apply what I had learned. This study has granted me the opportunity of fully understanding how to apply numerical methods and also the opportunity of working with some of the best technology centres in this field.

For this reason I am very satisfied of the work done. I can now say that I can develop a code that implements and solves many of the most common differential equations that are present in fluid dynamics as well as heat conduction, and I am confident that I can apply this to other differential equations if the need arises.

Conclusions of the results yield a very positive balance: four of the five cases have been completely solved and checked, and the last case could be addressed with a turbulent solver. I would have liked to design a code for the turbulent solution, but unfortunately the time is limited. Maybe in a future project this can be addressed.

## Recommendations

This study can be easily be expanded for an end of master project. The number of possibilities that can be developed is uncountable.

For instance a more extensive study of turbulence would be very advisable. The flux around the square cylinder cannot be simulated for Reynolds numbers higher than a very limited value. With a 2D turbulence solver the same problem can be solved for any Reynolds value, including the unsteady solution of the problem.

Another advisable modification of the code would be to adapt it for unstructured meshes. For non-square objects it would offer a better mesh adaptation. There are many programs that design non structured meshes that are extremely well designed and would offer a better solution. The alternative to this method would be to use the blocking-off method, which consists in creating a mesh which is so dense that the object is well defined by it.



Unfortunately this method is very expensive in terms of computational cost and is therefore less advisable.

Another modification that would greatly increase the number of applications of the code would be to implement a compressible Navier-Stokes solver. With this evolution the code would be able to simulate the flux of any fluid with the given geometry instead of being limited to compressible fluids. This modification is clearly out of the scope of an end of degree project, and would imply very important modifications to the code.

Finally an application could be designed, for instance Matlab offers a wide variety of options for designing applications that are very fun to use and offer very good results. Using for instance a graphical user interface does not add a very significant difficulty and enhances user experience with the program.

# Bibliography

[1] <http://www.airbus.com/presscentre/pressreleases/press-release-detail/detail/applying-new-technology-to-the-a350-xwb039s-development/> consulted 12/05/2015

All the CTTC problems and dossiers are available online for engineering students that are developing the end of degree project under the tuition of this institution. The following links include the problems and the specific information necessary to solve them:

[2] Laminar problems: <https://www.cttc.upc.edu/public/courses/numerical-methods-cfd> last consulted 20/05/2015

[3] Turbulence problems: <https://www.cttc.upc.edu/public/courses/turbulence> last consulted 20/05/2015

[4] Trias, F. X., Gorobets, a., Soria, M., & Oliva, a. (2010). Direct numerical simulation of a differentially heated cavity of aspect ratio 4 with Rayleigh numbers up to 10<sup>11</sup> - Part I: Numerical methods and time-averaged flow. International Journal of Heat and Mass Transfer, 53(4), 665–673. doi:10.1016/j.ijheatmasstransfer.2009.10.026

[5] M. Potter, D.C. Wiggert (2008), Fluid Mechanics (Schaum's Series), Schaum's Outlines, McGraw-Hill (USA), ISBN 978-0-07-148781-8

[6] B. E. Launder, D. B. Spalding (1974). "The Numerical Computation of Turbulent Flows". Computer Methods in Applied Mechanics and Engineering 3 (2): 269–289. DOI: 10.1016/0045-7825(74)90029-2

[7] J.N. Reddy (2006). An Introduction to the Finite Element Method (Third ed.). McGraw-Hill. ISBN 9780071267618.

[8] LeVeque, Randall (2002), Finite Volume Methods for Hyperbolic Problems, Cambridge University Press. ISBN: 9780521009249

[9] John Strikwerda (2004). Finite Difference Schemes and Partial Differential Equations (2nd ed.). SIAM. ISBN 978-0-89871-639-9.

[10] Breuer, M., Bernsdorf, J., Zeiser, T., & Durst, F. (2000). 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, 21(2), 186–196. doi:10.1016/S0142-727X(99)00081-8

[11] [http://archive.unu.edu/update/archive/issue31\\_5.htm](http://archive.unu.edu/update/archive/issue31_5.htm) consulted 24/05/2015