Master's Thesis

Master's Degree in Space and Aeronautical Engineering

FROM PURE CONDUCTION TO HOMOGENEOUS ISOTROPIC TURBULENCE

Àdel Alsalti Baldellou

Director: Dr. Carles-David Pérez Segarra
Co-Director: Dr. Asensi Oliva Llena

Heat Tranfer Center (CTTC)
ESEIAAT
Polytechnic University of Catalonia

Barcelona, January 2019

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

Contents

Abstract

The aim of this work is to study multiple cases related to fluid dynamics and heat transfer. More concretely, six cases of increasing complexity will be approached from the perspective of computational fluid dynamics, solving the partial differential equations related to each phenomenon by means of the finite volume method. The cases that have been simulated are: pure conduction, convection-diffusion, lid-driven cavity, differentially heated cavity, one and two-dimensional Burgers equation and homogeneous isotropic turbulence (implementing the Smagorinsky turbulence model).

Abstract

L'objectiu d'aquest treball és estudiar multiples casos relacionats amb la dinàmica de fluids i amb la transferència de calor. Més concretament, sis casos de complexitat creixent han estat abordats des de la perspectiva de la dinàmica de fluids computacional, resolent les equacions en derivades parcials associades a cada fenomen a partir del mètode de volums finits. Els casos que han estat simulats són: conducció pura, convecció-difusió, *lid-driven cavity*, *differentially heated cavity*, equació de Burgers en una i dues dimensions i turbulència isotròpica homogènia (implementant el model de turbulència de Smagorinsky).

Acknowledgements

I would like to specially thank the director and co-director of my master's thesis, Dr. Carles-David Pérez Segarra and Dr. Asensi Oliva Llena, for their implication and assistance.

Additionally, I would like to thank Jordi Chiva and Dr. Francesc Xavier Trias for their valuable comments and altruistic help throughout the elaboration of this work.

Finally, I would also like to show my gratitude to my family and friends for their unconditional support.

List of Figures

List of Tables

Introduction

In the present work there have been simulated six cases of increasing complexity involving fluid dynamics and heat transfer. With that aim, we have divided the text into three parts: firstly, there will be succinctly introduced all mathematical notions, mainly related to partial differential equations and finite volume method. Secondly, where the different cases are presented under the same structure, firstly showing the discretization of its governing PDEs and, finally, presenting the results obtained after implementing the discretized equations into separate C++ programs. Finally, some final remarks are made and possible lines of research are commented.

When it comes to the cases studied, all of them are rather iconic within the field of computational fluid dynamics (CFD). In particular, we are going to study the following phenomena:

- **Pure conduction**: Pure conduction within a square plate composed by 4 different materials.
- **Convection-diffusion**: Resolution of energy equation to obtain the distributions of temperature within two-dimensional fluids for different velocity fields.
- **Lid-driven cavity**: Resolution of Navier-Stokes momentum equation to obtain the velocity fields within a square fluid with fixed velocity at the top border.
- **Differentially heated cavity**: Coupled resolution of energy and momentum equations considering the approximation of Boussinesq for buoyancy.
- **Burgers equation**: Resolution of a simplification of Navier-Stokes momentum equation to get introduced into turbulence and its paradigmatic behaviour.
- **Homogeneous isotropic turbulence**: Combination of multiple numerical techniques to simulate homogeneous isotropic turbulence by approximating periodic solutions of Navier-Stokes equations.

All along the study of these different cases, different numerical methods to simulate them will be discussed and the validity of the results will be contrasted.

It is worth to mention that the successful simulation of homogeneous isotropic turbulence is due to the wise combination of our highly efficient resolution of Burgers equation by means of pseudo-spectral methods together and the fractional step method introduced in the study of lid-driven and differentially heated cavities. Additionally, the Smagorinsky model has been implemented in order to improve our simulations with coarse meshes.

In order to achieve all these results, there has been performed an intensive research on numerical methods, computational fluid dynamics and on C++ and MPI languages.

FFTW, a highly optimized open-source library, has been used to compute fast Fourier transforms and applied in all the problems involving spectral methods, namely Burgers equation and homogeneous isotropic turbulence.

The applications of the codes that has been implemented are really broad, given that the boundary conditions of each case can be easily updated, allowing the user to numerically calculate temperature, velocity and pressure fields. Additionally, the comparison in terms of efficiency of different numerical methods can be very useful to decide what sort of implementation develop. In particular, in the resolution of Burgers equation there has been shown the huge improvement that pseudo-spectral methods suppose with respect to spectral when non-linear terms are present.

Part I

Fundamentals of numerical analysis

The need for approximations

Most physical phenomena are governed by partial differential equations (PDEs). As a consequence, very high complexity is added to its understanding and prediction, given that the general analytical solution of most of the PDEs remain unknown.

Among many other, examples of very relevant PDEs in physics may be the following:

Laplace equation: $\Delta\phi = 0$

Schrödinger equation: $-\frac{\hbar^2}{2m}\Delta\phi + V\phi = i\hbar\frac{\partial\phi}{\partial t}$

Maxwell-Faraday equation: $\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$

Wave equation: $\frac{\partial^2\phi}{\partial^2 t} = c^2\Delta\phi$

However, in the present work we will only focus on PDEs related to incompressible fluid dynamics and heat and mass transfer, such as the following:

Convection-diffusion equation: $\frac{\partial(\rho\phi)}{\partial t} + \nabla\cdot(\rho\phi\mathbf{v}) = \nabla\cdot(\Gamma\nabla\phi) + S$

Continuity equation: $\frac{\partial\rho}{\partial t} = -\nabla\cdot(\rho\mathbf{v})$

Incompressible Navier-Stokes equation: $\frac{\partial\mathbf{v}}{\partial t} + (\mathbf{v}\cdot\nabla)\mathbf{v} = \nu\Delta\mathbf{v} - \frac{1}{\rho}\nabla p$

Among them, Navier-Stokes equation plays a fundamental role in fluid mechanics and, in spite of this fact, its analytical solution remains an open problem. This clearly shows the need to find a way to deal with such equations in order to be able to continue doing research in those fields governed by PDEs whose solutions have not been fully described.

For all these, scientists from many different branches have needed to find ways to approximate the solutions of their PDEs by means of a wide variety of numerical methods and making use of powerful supercomputers.

Throughout the following chapters of Part I, there will be introduced all the formalism used to execute these approximations following the same formal deduction made by S. Patankar in [1].

In Part II, all these results and, in particular finite volume method (FVM), will be applied intensively in the six different cases being studied.

Discretization of the space

The first step to tackle the problem of approximating solutions of PDEs consists of discretizing the physical domain that is intended to be studied. This procedure is of extreme importance, given that it will allow us to make our problem become discrete instead of continuous, and, as we will see in the following sections, it will allow us to approach it from an algebraic perspective.

The basic idea shared by many numerical methods to discretize the physical domain requires considering a set of nodes where we will calculate the physical properties that we may be studying. Therefore, if we are studying a square material, the simplest choice will be a set of nodes distributed homogeneously all along the plate as shown in Figure 0.1.

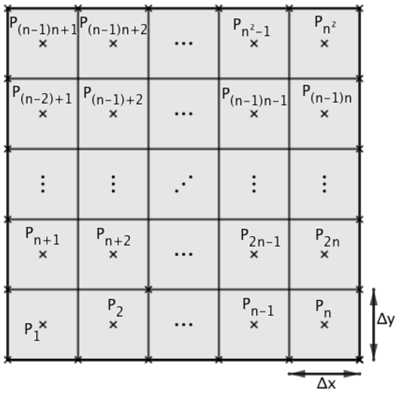


Figure 0.1: Uniform square mesh.

The grid resulting from dividing the physical domain is commonly referred as *mesh* and, as can be expected, gives rise to different results depending on the number of nodes it has and its distribution. This is why the discretization of the geometry is crucial in order to obtain reasonable results without wasting too much computational energy. To do so, we will have to consider enough nodes to reach the desired resolution, but trying to keep its number as low as possible to save as much computational energy as possible. A wise strategy could be, for example, the consideration of inhomogeneous meshes with higher concentration of nodes where more abrupt fluctuations of the studied field take place.

Finite volume method

At this point, we are ready to introduce the FVM. Given a mesh related to a certain physical domain, FVM will consider three-dimensional volumes (surfaces or segments in 2D or 1D, respectively) around the nodes of the mesh. This geometrical entities will be referred as *control volumes* (CV), and the fields being studied will be assumed to remain constant all across them.

Let us note that this could be defined differently, for example, assuming a linear variation of the field between adjacent nodes. Nevertheless, as we have already said, we will consider a stepwise profile of the field within the CVs. A reason for that, as noted in [1], resides in the physically coherent requirement of an integral mass, energy and momentum conservation over the whole system regardless of the mesh under consideration. This is ensured by imposing that fluxes at both sides of each face give the same result. Therefore, not all possible profiles of the field within the CVs would be acceptable. We just need to take a glance at Figure 0.2.

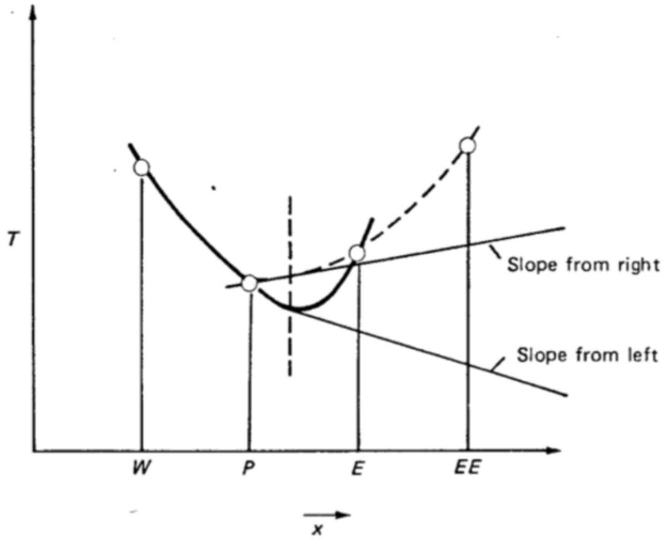


Figure 0.2: Image extracted from [1]. It shows four nodes belonging to a one-dimensional mesh and how the consideration of a quadratic field profile within the CVs results into a mismatch on the derivatives of the field calculated on each side of the face bordering nodes P and E.

Finally, it is worth to mention that there exist other alternatives to the FVM, such as the finite element method or the finite difference method, which are out of the scope of this work.

Discretization of PDEs

In this section we will give general guidelines to understand how differential operators are approximated within the formalism of FVM, giving rise to the discretized PDE. In spite of the fact that, as we already said, all the deductions of this part are based on the work made by S. Patankar in [1], there has been also revised the more detailed study developed by C.D. Pérez-Segarra et al. in [2].

Hence, two main topics will be discussed: discrtized differential operators and temporal integration schemes.

Spatial differential operators

Given the wide variety of existing approximations of differential operators, this subsection could fill a complete independent work. However, pragmatism will be imposed and we will just introduce the ones that have been used in our simulations. More particularly, we will make use of a first order approximation over a regular mesh. To do so, let us consider the following Taylor expansions of a field ϕ:

$$\left\{ \begin{array}{l} \phi_E = \phi_e + \left.\dfrac{\partial \phi}{\partial r}\right|_e \dfrac{\Delta r}{2} + O(\Delta r)^2 \\ \phi_P = \phi_e + \left.\dfrac{\partial \phi}{\partial r}\right|_e \dfrac{\Delta r}{2} + O(\Delta r)^2 \end{array} \right. \implies \left.\frac{\partial \phi}{\partial r}\right|_e \simeq \frac{\phi_E - \phi_P}{\Delta r}, \tag{1}$$

where P stands for the node related to the CV we are working with, E for the node corresponding to the adjacent cell in the positive axial direction, e for the surface between both CVs and Δr for the distance between P and E.

Therefore, partial derivatives with respect to any direction r become clearly defined and, consequently, also the rest of differential operators such as the gradient or the divergence, as they can be immediately obtained from the spatial partial derivatives of the field.

Temporal integration schemes

Exactly as we did with the physical domain, the simulated temporal interval can be divided into discrete pieces known as *time-steps*. In transient processes, this procedure is as necessary as the discretization of the space, given that it allows us to calculate partial derivatives with respect to time and to perform temporal integrals.

Analogously to what we did in Equation 1, we could approximate temporal derivatives as follows:

$$\left.\frac{\partial \phi}{\partial t}\right|_{t_n} \simeq \frac{\phi_P^{n+1} - \phi_P^n}{t_{n+1} - t_n} \tag{2}$$

Additionally, further assumptions need to be made in order to compute the temporal integral of any function depending on time (implicitly or explicitly), $g(t)$. More concretely, exactly as we did for the distribution of ϕ within a CV, we have to set the distribution of $g(t)$ within a time-step, $\Delta t = t_{n+1} - t_n$. Although there are several options available for that, we have summarized the main ones in Figure 0.3.

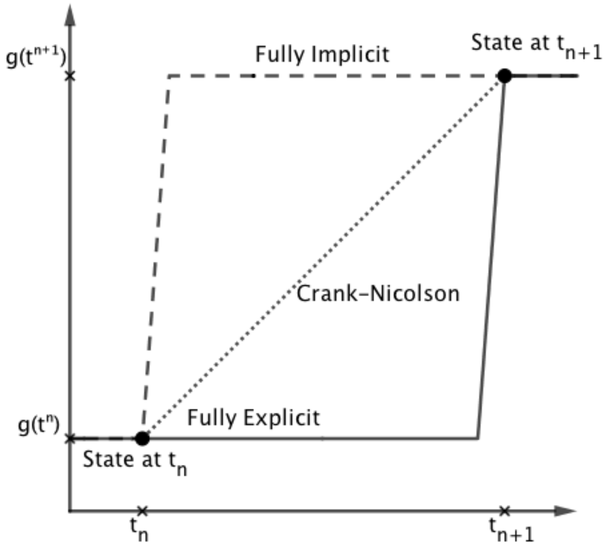


Figure 0.3: Image extracted from [1]. It shows three temporal schemes: fully explicit, fully implicit and Crank-Nicolson approaches to the temporal evolution of functions between consecutive time-steps.

The first one (fully explicit) assumes that g does not change and remains constant at the value corresponding to t_n until t_{n+1}, when it suddenly becomes $g(t^{n+1})$. The second option (Crank-Nicolson) consists of assuming that the temperature varies linearly between every time-step. Finally, the fully implicit assumption, assumes that immediately after t_n, g reaches the value corresponding to the next time-step, $g(t^{n+1})$.

Therefore, we would obtain the following results depending on the temporal scheme under consideration:

$$\begin{cases} \int_{t_n}^{t_{n+1}} g(t)dt \simeq g(t^n)\Delta t, & \text{for fully explicit scheme} \\ \int_{t_n}^{t_{n+1}} g(t)dt \simeq \dfrac{g(t^{n+1}) - g(t^n)}{2}\Delta t, & \text{for Crank-Nicolson scheme} \\ \int_{t_n}^{t_{n+1}} g(t)dt \simeq g(t^{n+1})\Delta t, & \text{for fully implicit scheme} \end{cases} \tag{3}$$

or, more generally, we could write:

$$\int_{t_n}^{t_{n+1}} g(t)dt \simeq \left(\beta g(t^n) + (1-\beta) g(t^{n+1})\right)\Delta t, \text{ where } \beta \in [0, 1].$$

Part II

Numerical simulations: from pure conduction to turbulence

Chapter 1

Pure conduction

In this chapter we will simulate the heat transfer within a square plate composed by 4 different materials as shown in the figure below, starting from a uniform distribution equal to $T_0 = 11^{\underline{\text{o}}}C$.

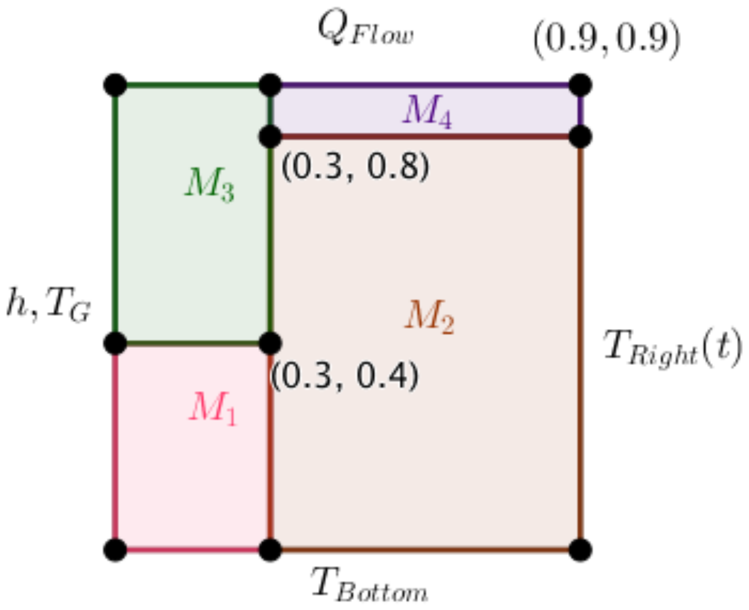


Figure 1.1: Distribution of the different materials

The physical properties of each material are summarized in Table 1.1.

Material	ρ ($^{kg}/_{m^3}$)	c_p ($^{J}/_{kgK}$)	k ($^{W}/_{mK}$)
M_1	2500.0	970.0	180.0
M_2	2700.0	930.0	140.0
M_3	2200.0	710.0	150.0
M_4	1700.0	920.0	140.0

Table 1.1: Physical properties of each material.

Additionally, at the borders of the plate there have been set the boundary conditions shown in Table 1.2.

Material	**Boundary condition**
Bottom	$T = 18.0^{\underline{o}}C$
Top	$Q_{flow} = 89.0^{W}/_{m}$
Left	$T_g = 35.0^{\underline{o}}C$ and $\alpha_g = 8.0^{W}/_{m^2K}$
Right	$T(t) = 11.0 + 0.006t^{\underline{o}}C$

Table 1.2: Physical properties of each material.

Thus, there will be implemented a C++ program based on FVM as presented in [1] by S. Patankar.

1.1 Discretization

The case under study is an example of unsteady two-dimensional heat conduction, which is governed by the diffusion equation:

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T)$$

Integrating it over the surface of a sufficiently small two-dimensional CV, S, would result into the following equation:

$$\int_S \rho c_p \frac{\partial T}{\partial t} dA = \int_S \nabla \cdot (k \nabla T) dA \tag{1.1}$$

If we remember that the material is identical throughout S, as well as its temperature, and if we make use of the divergence theorem, we could rewrite Equation 1.1 as:

$$\rho c_p \frac{\partial T}{\partial t} S = k \oint_C \nabla T \cdot \hat{n} ds \tag{1.2}$$

Additionally, given the regular square structure of the mesh that we are using, we can easily calculate the integral at the right-side of Equation 1.2:

$$\rho c_p \frac{\partial T}{\partial t} S = \sum_{\text{faces of } S} k_f \left(\left.\frac{\partial T}{\partial x}\right|_f, \left.\frac{\partial T}{\partial y}\right|_f \right) \cdot (n_x, n_y)\, L_f \tag{1.3}$$

where L_f refers to the length of the face (either Δx or Δy). Furthermore, if we approximate the gradient of T in terms of the neighbours of a generic point P, we obtain the following equation:

$$\begin{aligned} \rho c_p \frac{\partial T}{\partial t} S \simeq{}& k_N \left(\left.\frac{\partial T}{\partial x}\right|_N, \frac{T_N - T_P}{\delta y} \right) \cdot (0,1)\Delta x + k_S \left(\left.\frac{\partial T}{\partial x}\right|_S, \frac{T_P - T_S}{\delta y} \right) \cdot (0,-1)\Delta x \\ &+ k_E \left(\frac{T_E - T_P}{\delta x}, \left.\frac{\partial T}{\partial y}\right|_E \right) \cdot (1,0)\Delta y + k_W \left(\frac{T_P - T_W}{\delta x}, \left.\frac{\partial T}{\partial y}\right|_W \right) \cdot (-1,0)\Delta y, \end{aligned} \tag{1.4}$$

where δx and δy make reference to the distance between a point and the closest P_i following either the north, south, east or west direction. However, as we will see in Section 1.1.1, this will not be totally true, given that it will only hold for the points inside the plate, and specific considerations will be needed for all the points located at the borders of the plate.

Thus, it immediately follows from Equation 1.4 that:

$$\rho c_p \frac{\partial T}{\partial t} S = k_N \frac{T_N - T_P}{\delta y} \Delta x + k_S \frac{T_S - T_P}{\delta y} \Delta x + k_E \frac{T_E - T_P}{\delta x} \Delta y + k_W \frac{T_W - T_P}{\delta x} \Delta y \tag{1.5}$$

Let us note that the right-member of the Equation 1.5 is a function depending on the temperature. Consequently, we could rewrite it as:

$$\rho c_p \frac{\partial T}{\partial t} S = g(T) \tag{1.6}$$

Therefore, if we integrate Equation 1.6 between two consecutive time-steps, we reach:

$$\int_{t_n}^{t_{n+1}} \rho c_p \frac{\partial T}{\partial t} S dt = \int_{t_n}^{t_{n+1}} g(T) dt \tag{1.7}$$

On the one hand, bearing in mind that time-steps are small enough, we can conclude that the left-hand side of Equation 1.7 satisfies:

$$\int_{t_n}^{t_{n+1}} \rho c_p \frac{\partial T}{\partial t} S dt \simeq \rho c_p (T_P^{n+1} - T_P^n) S \tag{1.8}$$

On the other hand, considering a fully implicit scheme (as presented in Part I), we can assert that:

$$\int_{t_n}^{t_{n+1}} g(T) dt \simeq g(T^{n+1}) \Delta t \tag{1.9}$$

Finally, if we combine Equation 1.7 with the equalities 1.8 and 1.9, we obtain the following:

$$\begin{aligned} \frac{\rho c_p (T_P^{n+1} - T_P^n) S}{\Delta t} &= k_N \frac{T_N^{n+1} - T_P^{n+1}}{\delta y} \Delta x + k_S \frac{T_S^{n+1} - T_P^{n+1}}{\delta y} \Delta x \\ &+ k_E \frac{T_E^{n+1} - T_P^{n+1}}{\delta x} \Delta y + k_W \frac{T_W^{n+1} - T_P^{n+1}}{\delta x} \Delta y, \end{aligned}$$

which can be rewritten in its final discretized form as:

$$a_P T_P^{n+1} = a_N T_N^{n+1} + a_S T_S^{n+1} + a_E T_E^{n+1} + a_W T_W^{n+1} + b, \tag{1.10}$$

where the coefficients a_P, a_N, a_S, a_E, a_W and b are defined as follows:

$$\begin{aligned} a_P &= \tfrac{k_N \Delta x}{\delta y} + \tfrac{k_S \Delta x}{\delta y} + \tfrac{k_E \Delta y}{\delta x} + \tfrac{k_W \Delta y}{\delta x} + \tfrac{\rho c_p S}{\Delta t} \\ a_N &= \tfrac{k_N \Delta x}{\delta y} \\ a_S &= \tfrac{k_S \Delta x}{\delta y} \\ a_E &= \tfrac{k_E \Delta y}{\delta x} \\ a_W &= \tfrac{k_W \Delta y}{\delta x} \\ b &= \tfrac{\rho c_p S T_P^n}{\Delta t} \end{aligned} \tag{1.11}$$

Thus, we have finally obtained the system of equations that we are going to solve recursively. Starting with a uniform temperature throughout the plate of $11^{\text{o}}C$, we will be able to calculate the distribution of temperatures at t_1. After that, we will be able to solve the equations for t_2, then for t_3, and so on.

1.1.1 Boundary Conditions

Even thought we have already found the system of equations that has to be solved in order to predict the heat transference within our plate, we still need to find the way of implementing the boundary conditions that define the situation presented in the problem. Actually, this is how we are going to be able to find its unique solution.

First boundary condition to be considered, and perhaps the simplest one, is to set the initial temperature of the whole plate at $T_0 = 11^{\text{o}}C$. This can be trivially done by setting $T_P = T_0$, for all the points of the plate at $t = 0$. Thus, when we try to calculate T_P^{n+1}, which from now on we are going to call T_P, we will always have properly defined the temperature corresponding to the previous time-step, T_P^n.

Top border - Constant Heat Flow, Q_{Flow}

Recalling the Fourier's law, we can see that the vertical heat flow coming from the top border of the plate will satisfy the following formula:

$$|Q_{Flow}| = k\frac{\partial T}{\partial y}$$

that could be approximated as (see Figure 1.2):

$$|Q_{Flow}| \simeq k_N\frac{T_N - T_P}{\delta y}$$

However, let us point that, as we introduced in the previous section and can be easily seen in the image below, in this case we will have that $\delta y = \frac{\Delta y}{2}$.

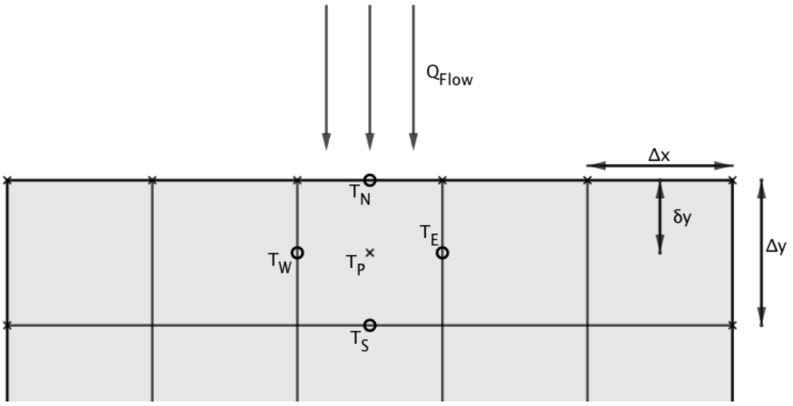


Figure 1.2: Schema of the boundary condition at the top border.

Hence, recalling the term of Equation 1.10,

$$k_N\frac{T_N - T_P}{\delta y}\Delta x = Q_{Flow}\Delta x$$

and so, we will have that:

$$a_PT_P = a_ST_S + a_ET_E + a_WT_W + b, \tag{1.12}$$

with:

$$\begin{aligned} a_P &= \tfrac{k_S\Delta x}{\Delta y} + \tfrac{k_E\Delta y}{\Delta x} + \tfrac{k_W\Delta y}{\Delta x} + \tfrac{\rho c_p \Delta x\Delta y}{\Delta t} \\ a_S &= \tfrac{k_S\Delta x}{\Delta y} \\ a_E &= \tfrac{k_E\Delta y}{\Delta x} \\ a_W &= \tfrac{k_W\Delta y}{\Delta x} \\ b &= \tfrac{\rho c_p S T_P^n}{\Delta t} + Q_{Flow}\Delta x \end{aligned} \tag{1.13}$$

Left border - Gas Convection, h and T_G

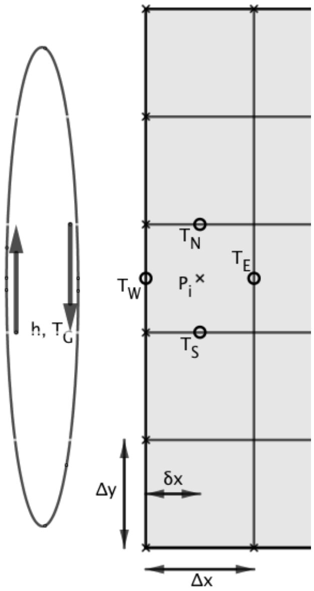


Figure 1.3: Schema of the boundary condition at the left border.

Similarly to what we saw at the top border, at the left one we also have a heat flow. Nevertheless, in this case it is due to the convection with a gas with a determined heat transfer coefficient, h, and temperature, T_G.

Let us consider a point, P_i, located at the left border of the plate (as shown in Figure 1.3). As we did in the precious case, we are going to relate it with the temperature of its neighbours.

Considering the law of convective heat transfer, we can reach the following equality:

$$|Q_{Left}| = h(T_G - T_W)$$

and by means of Fourier's law, we can assert that:

$$h(T_G - T_W) = k\frac{\partial T}{\partial x}$$

Thus, as we did before, we can approximate it as (see Figure 1.3):

$$h(T_G - T_W) \simeq k_N \frac{T_W - T_P}{\delta x} \tag{1.14}$$

where, similarly to what happened before, $\delta x = \frac{\Delta x}{2}$.

It immediately follows from Equation 1.14 that:

$$T_W = \frac{2k_W}{\Delta x(h + \frac{2k_W}{\Delta x})} T_P + \frac{hT_G}{h + \frac{2k_W}{\Delta x}} \tag{1.15}$$

and if we substitute this expression within Equation 1.10, we reach the following result:

$$a_P T_P = a_N T_N + a_S T_S + a_E T_E + b, \tag{1.16}$$

with:

$$\begin{array}{rcl}
a_P & = & \frac{k_N \Delta x}{\Delta y} + \frac{k_S \Delta x}{\Delta y} + \frac{k_E \Delta y}{\Delta x} + \frac{\rho c_p \Delta x \Delta y}{\Delta t} + \frac{k_W h \Delta y}{k_W + \Delta x h} \\
a_N & = & \frac{k_N \Delta x}{\Delta y} \\
a_S & = & \frac{k_S \Delta x}{\Delta y} \\
a_E & = & \frac{k_E \Delta y}{\Delta x} \\
b & = & \frac{\rho c_p S T_P^n}{\Delta t} + \frac{k_W h \Delta y T_G}{k_W + \Delta x h}
\end{array}$$

Lower border - Isotherm, T_{Bottom}

Given a point P_i such as the one shown in Figure 1.4, we will just have to impose that $T_P = T_{Bottom}$. Thus, we will reach the following results:

$$a_P T_P = a_N T_N + a_E T_E + a_W T_W + b,$$

with:

$$\begin{aligned}
a_P &= \frac{k_N \Delta x}{\Delta y} + \frac{2k_S \Delta x}{\Delta y} + \frac{k_E \Delta y}{\Delta x} + \frac{k_W \Delta y}{\Delta x} + \frac{\rho c_p \Delta x \Delta y}{\Delta t} \\
a_N &= \frac{k_N \Delta x}{\Delta y} \\
a_E &= \frac{k_E \Delta y}{\Delta x} \\
a_W &= \frac{k_W \Delta y}{\Delta x} \\
b &= \frac{\rho c_p S T_P^n}{\Delta t} + \frac{2k_S \Delta x T_{Bottom}}{\Delta y}
\end{aligned}$$

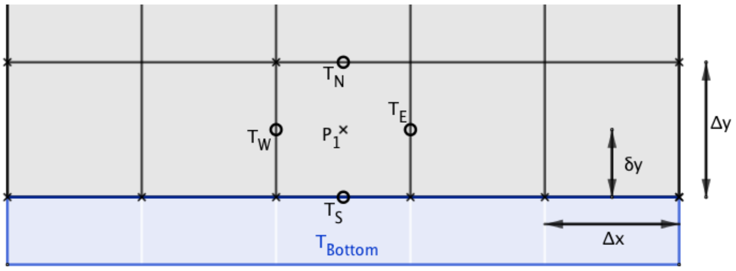


Figure 1.4: Schema of the boundary condition at the lower border.

Right border - Determined Function for Temperature, $T_{Right}(t)$

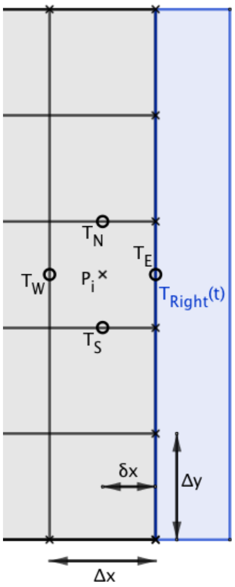


Figure 1.5: Schema of the boundary condition at the right border.

Finally, given a point P_i, such as the one shown in Figure 1.5, we will just have to proceed exactly as we did for the previous case but bearing in mind that now, instead of T_{Bottom}, we will have a function, $T_{Right}(t)$, depending on time. Thus, we will just have to impose that $T_E = T_{Right}(t)$ and we will reach the following results straightforwardly:

$$a_P T_P = a_N T_N + a_S T_S + a_W T_W + b,$$

with:

$$\begin{aligned}
a_P &= \frac{k_N \Delta x}{\Delta y} + \frac{k_S \Delta x}{\Delta y} + \frac{2k_E \Delta y}{\Delta x} + \frac{k_W \Delta y}{\Delta x} + \frac{\rho c_p \Delta x \Delta y}{\Delta t} \\
a_N &= \frac{k_N \Delta x}{\Delta y} \\
a_S &= \frac{k_S \Delta x}{\Delta y} \\
a_W &= \frac{k_W \Delta y}{\Delta x} \\
b &= \frac{\rho c_p S T_P^n}{\Delta t} + \frac{2k_E \Delta y T_{Right}(t)}{\Delta x}
\end{aligned}$$

1.1.2 Material Interfaces

Finally, the last detail to be mentioned, is the treatment that we are making of the points, P_i, such that any of their faces is in contact with more than one different materials.

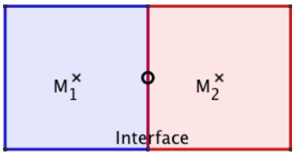


Figure 1.6: Example of face in contact with two different materials.

It is clear that some special care should be taken with the conductivity of such faces. It would be reasonable to consider the conductivity of the interface as the arithmetic mean of k_1 and k_2. However, this would lead to an undesired result: if, for example, $k_1 = 0$ and $k_2 > 0$, then $k_{Interface} > 0$, which would not make any sense, given that M_1 is an ideal thermal insulator.

In order to avoid this behaviour, the harmonic mean could be used:

$$k_{Interface} = \frac{2k_1k_2}{k_1 + k_2}$$

However, a more formal justification of this choice could be given. Bearing in mind the balance that must exist at the interface on the heat fluxes, we could write the following equality:

$$\dot{q}_e = \dot{q}_e^- = \dot{q}_e^+ \implies \dot{q}_e = -k_1\frac{T_e - T_P}{(\delta x)_{Pe}} = -k_2\frac{T_E - T_e}{d_{eE}}$$

Therefore,

$$\begin{cases} \dfrac{\dot{q}_e(\delta x)_{Pe}}{k_1} = -(T_e - T_P) \\ \dfrac{\dot{q}_e(\delta x)_{eE}}{k_2} = -(T_E - T_e) \end{cases} \implies \dot{q}_e = -\frac{(\delta x)_{PE}}{\frac{(\delta x)_{Pe}}{k_1} + \frac{(\delta x)_{eE}}{k_2}}\frac{T_E - T_P}{(\delta x)_{PE}},$$

which, in case of considering a regular mesh becomes:

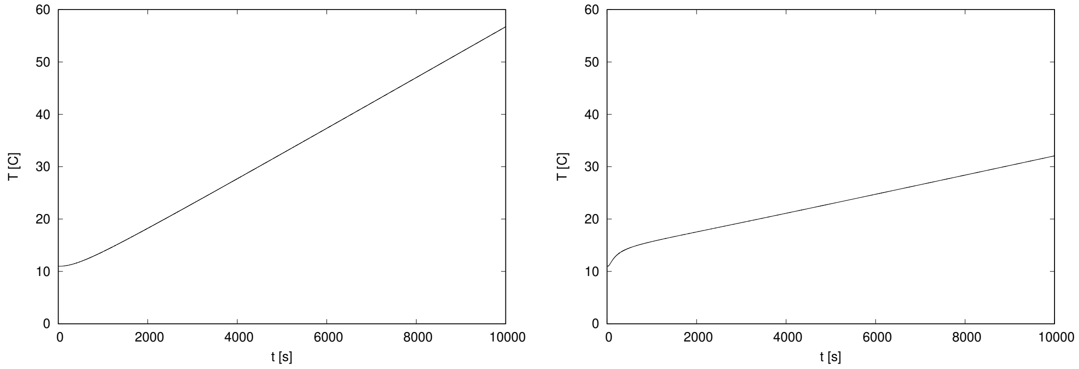
$$\dot{q}_e = -\frac{2k_1k_2}{k_1 + k_2}\frac{T_E - T_P}{\Delta x}, \tag{1.17}$$

providing complete coherence to the election of $k_{Interface}$.

1.2 Results

Finally, we have created a C++ program that runs all the calculations defined so far, starting from $t = 0$ until $t = 10000s$.

In the following two charts, there is shown the temporal evolution of the temperature at distinct points of the plate:

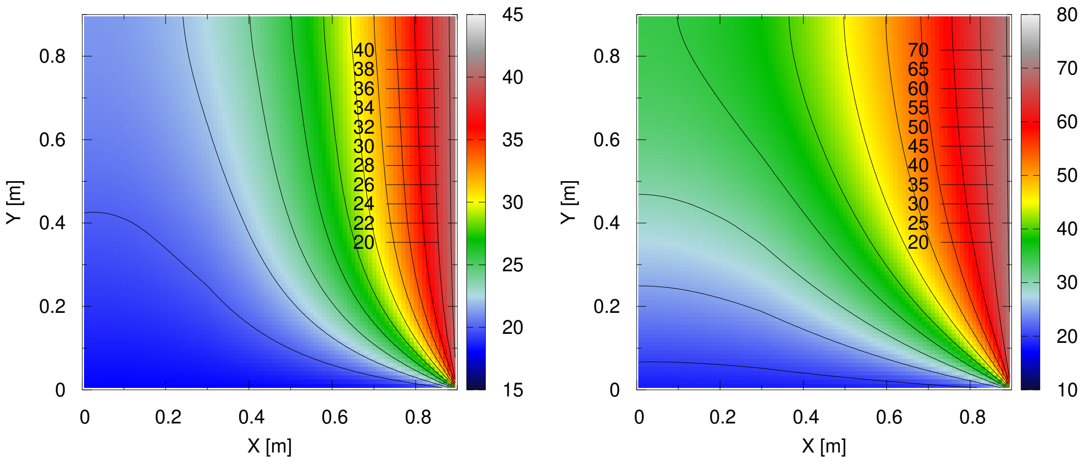


(a) $(x, y) = (0.73, 0.62)$

(b) $(x, y) = (0.60, 0.16)$

Figure 1.7: Evolution of the temperature at two points of the plate.

More generally, the following heat-maps show the distribution of the temperature throughout the plate at the middle of the temporal interval to be simulated and at its end:



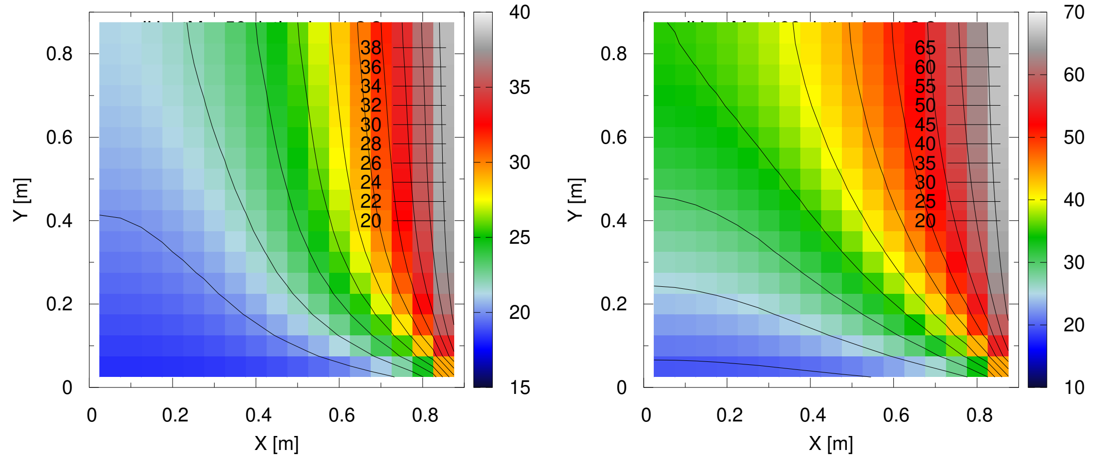
(a) $t = 5000s$

(b) $t = 10000s$

Figure 1.8: Distribution of the temperature throughout the plate at different instants of time.

At this point, it is interesting to make a little study regarding the effect of the mesh on the results of the simulation. With this aim, we have run the same code using other values of n apart from $n = 90$. Obviously, a lower n will mean a worse result, whereas a bigger one will lead to more precision (and more time of computation).

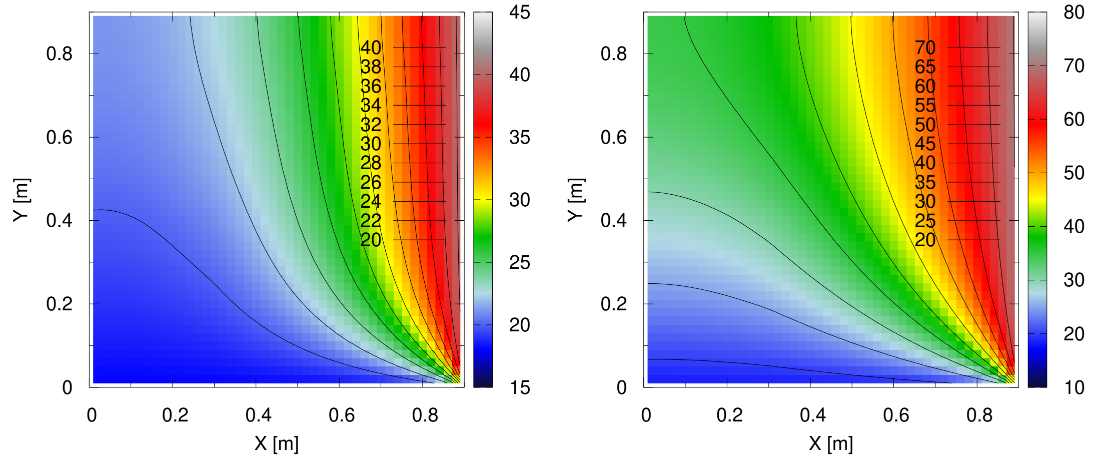
On the images below there is shown the distribution of the temperature throughout the plate for $n = 18$, $n = 45$ and $n = 180$. It is worth to mention that we have not included the graphics showing the evolution of the temperature at $(x, y) = (0.73, 0.62)$ and $(x, y) = (0.60, 0.16)$, because they are so similar that its differences would not be noticeable.



(a) $t = 5000s$ and $n = 18$

(b) $t = 10000s$ and $n = 18$

Figure 1.9: Distribution of the temperature throughout the plate at different instants of time using $n = 18$.



(a) $t = 5000s$ and $n = 45$

(b) $t = 10000s$ and $n = 45$

Figure 1.10: Distribution of the temperature throughout the plate at different instants of time using $n = 45$.

Finally, it is worth to go back a few steps and remember when we decided, without any justification, to utilize the fully implicit schema (see Figure 0.3). Further comments about that choice should definitely be made.

On the one side, let us note that if the temperature of either T_N, T_S, T_E or T_W increases, so should do T_P (this is an obvious consequence of conductive heat transference). Thus, any physically realistic solution that we obtain should satisfy it. Therefore, if we consider Equation 1.10, it is clear that, in order to reach a coherent solution, all of its coefficients have to be positive (even b, given that it relates T_P^{n+1} with T_P^n).

On the other side, it can be seen that fully explicit and Crank-Nicolson schemas can lead to negative coefficients of the discretized equation for big enough values of Δt. Thus, the choice of an excessively big time-step would lead to unrealistic results. Nevertheless, the schema that we have considered (fully implicit), totally avoids this problematic, allowing us to consider wider time-steps and reach faster the stationary state (although losing resolution of the transient evolution).

Testing the same code with different values of Δt, result in surprisingly similar values of the temperature. For example, $T(x = 0.60, y = 0.16, t = 10000s)$ gives a result of $32.071^{\text{o}}C$ for Δt equal to $1s$, $2s$, $4s$ and $10s$. Only considering an extreme value of Δt such as $100s$ would result in a significantly different value equal to $32.0775^{\text{o}}C$. Another interesting feature to test is the difference of the time that running the whole simulation takes in a computer depending on the Δt under consideration. The following table summarizes several results:

Table 1.3: Comparison of the time that the same computer takes to run the whole simulation with respect to the Δt under consideration.

Δt	**Simulation time**
1s	3min 34s
2s	2min 56s
4s	2min 42s
10s	2min 33s
100s	1min 25s

Chapter 2

Convection-Diffusion

In this chapter we will simulate four different situations governed by the convection-diffusion equation, which can be written as follows:

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

All of the simulations will study the distribution of a general variable ϕ characterized by its diffusion coefficient, Γ, and will consider a two-dimensional material as the one shown in Figure 2.1. Additionally, the velocity field, $\vec{v}$, will be given in every case.

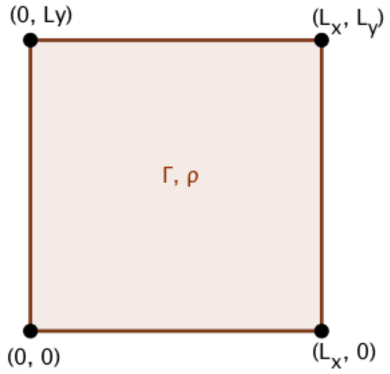


Figure 2.1: Schema of the plate amongst which the simulations are run.

In some cases it will be possible to find the analytical solution of the convection-diffusion equation. If so, it will be used to contrast the results obtained from the simulation.

Therefore, in the following sections there will be implemented a C++ program based on FVM following the same lines as S. Patankar in [1].

2.1 Discretization

The cases under study are examples of unsteady two-dimensional heat convection-diffusion, which, as we have already advanced, are governed by the following equation:

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho\phi\vec{v}) = \nabla \cdot (\Gamma\nabla\phi) + S \tag{2.1}$$

Since there is not either energy production or consumption in any of the cases we are dealing with, then we can assume $S = 0$. Thus, we have that:

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho\phi\vec{v} - \Gamma\nabla\phi) = 0 \tag{2.2}$$

It is possible to define the total heat flux as:

$$\vec{J} \equiv \overset{\text{convection}}{\rho\phi\vec{v}} - \overset{\text{diffusion}}{\Gamma\nabla\phi} . \tag{2.3}$$

Additionally, we can define the Peclet number as follows:

$$P_x = \frac{\rho v_x \Delta x}{\Gamma} \text{ and } P_y = \frac{\rho v_y \Delta y}{\Gamma}, \tag{2.4}$$

which represents a balance between the strength of convection with respect to diffusion. Therefore, fluids with a big Peclet number will present a predominance of convection in oposition to diffusion, and viceversa.

Thus, recalling Equation 2.3 we can write:

$$J_x^* \equiv \frac{J_x \Delta x}{\Gamma} = P_x\phi - \frac{d\phi}{d(x/\Delta x)} \text{ and } J_y^* \equiv \frac{J_y \Delta y}{\Gamma} = P_y\phi - \frac{d\phi}{d(y/\Delta y)}. \tag{2.5}$$

As seen by S. Patankar in [1], it is possible to see that the values of J_x^* and J_y^* at the interfaces between control volumes depend on the values of ϕ inside those control volumes. Aditionally, S. Patankar is able to see that this dependence is closely related to the Peclet number arriving to the following discretized form of Equation 2.2:

$$a_P\phi_P = a_N\phi_N + a_S\phi_S + a_E\phi_E + a_W\phi_W + b, \tag{2.6}$$

where the following definitions are given for the coefficients a_i and b:

$$\begin{array}{rcl} a_N &=& D_N A(|P|) + \max\{-F_N, 0\} \\ a_S &=& D_S A(|P|) + \max\{F_S, 0\} \\ a_E &=& D_E A(|P|) + \max\{-F_E, 0\} \\ a_W &=& D_W A(|P|) + \max\{F_W, 0\} \\ a_P^0 &=& \frac{\rho_P^0 \Delta x \Delta y}{\Delta t} \\ b &=& a_P^0 \phi_P^0 \\ a_P &=& a_N + a_S + a_E + a_W + a_p^0 \end{array} \tag{2.7}$$

with the following parameters:

$$D_N = \frac{\Gamma_N \Delta x}{(\delta y)_N},\ D_S = \frac{\Gamma_S \Delta x}{(\delta y)_S},\ D_E = \frac{\Gamma_E \Delta y}{(\delta x)_E} \text{ and } D_W = \frac{\Gamma_W \Delta y}{(\delta x)_W},$$

$$F_N = (\rho v_y)_N \Delta x,\ F_S = (\rho v_y)_S \Delta x,\ F_E = (\rho v_x)_E \Delta y \text{ and } F_W = (\rho v_x)_W \Delta y,$$

where δx and δy make reference to the distance between a point and the closest P_i following either the north, south, east or west direction. Additionally, it can be easily checked that: $P_i = {}^{F_i}/_{D_i}$.

Even more interestingly, S. Patankar is able to identify several schemes for the convection-diffusion models in terms of the function $A(|P|)$ that one considers. They are summarized in Table 2.1.

Table 2.1: Relationship of several schemes for the discretization of the convection-diffusion equation and its related function $A(|P|)$.

Scheme	$\mathbf{A(\|P\|)}$
Upwind	1
Central difference	$1 - 0.5\|P\|$
Hybrid	$\max\{1 - 0.5\|P\|, 0\}$
Power Law	$\max\left\{(1 - 0.1\|P\|)^5, 0\right\}$
Analytical solution	${}^{\|P\|}/_{(e^{\|P\|} - 1)}$

A high level analysis of each scheme would result really appropriate in order to justify the choice that we are going to make afterwards. It is obvious that, when it comes to choosing a scheme, the one that would give more exact results is the one corresponding to the analytical solution of the convection-diffusion equation. However, as the computing costs of working with an exponential function would be high, it makes sense to consider a simplification of that scheme that provides a good balance between quality of the results obtained and the computation resources that they require.

Firstly, the upwind scheme gives really unrealistic results in presence of a convection, therefore, it should be absolutely avoided. Next, we find the central difference scheme, which is not appropriate if considerable velocities are involved. However, the hybrid model, an improvement of the central difference one, works well in presence of either low or high velocities, but returning poor results if $|P| \in (1.5, 3.5)$. Finally, there is the power law scheme, which gives a really good approximation of the analytical solution involving less computing resources. For this reason, we have worked with this scheme, given that it represented a good balance of efficiency and exactitude.

It si worth to mention that further and more elaborated considerations related to the different schemes presented can be found in [3].

Thus, our implementation of the FVM applied to the convection-diffusion equation will be entirely based on the equations 2.6 and 2.7, considering $A(|P|) = \max\left\{(1 - 0.1|P|)^5, 0\right\}$.

2.1.1 Boundary Conditions

All over the last section, we have found the system of equations that has to be solved in order to predict the heat transference within our plate. Nevertheless, we still need to find the way of implementing the boundary conditions that define the different situations presented in the exercise. Actually, this is how we are going to be able to find their unique solutions.

First boundary condition to be considered, and perhaps the simplest one, is to set the initial value of ϕ at any point of the plate equal to ϕ_0. This can be trivially done by setting $\phi_P = \phi_0$, for all the points of the plate at $t = 0$. Thus, when we try to calculate ϕ_P^{n+1}, which from now on we are going to call ϕ_P, we will always have properly defined value of ϕ corresponding to the previous time-step, ϕ_P^n.

Two other boundary conditions will be consider throughout the different cases: fixed value of ϕ and null heat flow at the borders of the material. In the following two subsections we will show how to set each of these conditions.

Null Heat Flow

In order to show how to deal with a boundary condition that sets as zero the heat flux at some of the points located at the border of the material, we will initially assume that the boundary condition is applied to the top border of the plate, with the aim of generalizing the procedure afterwards.

Recalling the Fourier's law, we can see that the vertical heat flow coming from the top border of the plate will satisfy the following formula:

$$|Q_{Flow}| = \Gamma \frac{\partial \phi}{\partial y}$$

that could be approximated as (see Figure 1.2):

$$|Q_{Flow}| \simeq \Gamma_N \frac{\phi_N - \phi_P}{\delta y}.$$

Thus, if $|Q_{Flow}| = 0$, we will have that:

$$\phi_N = \phi_P.$$

Hence, we will have that:

$$(a_N + a_S + a_E + a_W + a_P^0)\phi_P = a_N\phi_P + a_S\phi_S + a_E\phi_E + a_W\phi_W + b,$$

and, consequently, the discretized equation will become:

$$(a_S + a_E + a_W + a_P^0)\phi_P = a_S\phi_S + a_E\phi_E + a_W\phi_W + b, \tag{2.8}$$

Note that, if we had to impose a heat flux equal to zero in any other of the borders of the material, we would actually reach the same results as in Equation 2.8, but suppressing the values corresponding to the appropriate neighbour instead of the ones of ϕ_N.

Fixed value of ϕ

Once again, with the aim of showing the procedure to impose a boundary condition that determines the value of ϕ at some of the points of the border of the material, we will firstly give an example assuming that ϕ is known at the right border, in order to generalize the procedure afterwards.

Hence, given a point P_i, such as the one shown in Figure 1.5, we will just have to impose that:

$$\phi_E = \phi_{Right}(t), \tag{2.9}$$

then, it immediately follows that:

$$a_P\phi_P = a_N\phi_N + a_S\phi_S + a_E\phi_{Right}(t) + a_W\phi_W + b. \tag{2.10}$$

Note that if we were given the value of ϕ at any other of the borders of the plate, we would proceed identically but adapting equations 2.9 and 2.10 to the corresponding neighbour of ϕ_P.

2.2 Results

Finally, we have created a C++ program that runs all the calculations defined so far, but adapting it to four different cases and their particular boundary conditions.

However, before presenting the different cases let us note that we have made use of the Power-law schema of $A(P)$ to run the simulations in spite of the fact that we could have used either the central difference, the hybrid or the exponential schemas, reaching with all of them really similar results.

2.2.1 Unidimensional flow with a unidimensional variation of the variable solved in the same direction of the flow

The first case corresponds to a rectangular material with an intern convection going in the x-direction with a constant velocity. The boundary conditions in the top and lower borders correspond to a null heat flux (see Section 2.1.1), and in the right and left borders to two different fixed values of ϕ (see Section 2.1.1). A diagram sumarizing this case can be found in the image below.

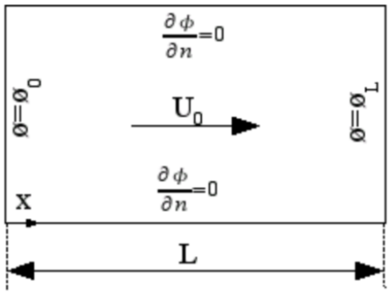

$\frac{\partial\phi}{\partial n}=0$
$\emptyset=\emptyset_0$
U_0
$\emptyset=\emptyset_L$
$\frac{\partial\phi}{\partial n}=0$
x
L

Figure 2.2: Schema of the boundary conditions for the case corresponding to the unidimensional flow with a unidimensional variation of the variable solved in the same direction of the flow.

Therefore, we have applied all the calculations defined so far to this particular case and run them until the instant when the time derivative of ϕ became smaller than 1.0e-6 (defining this as an acceptable approach to the stationary state). By doing so, we have obtained the distribution of ϕ all shown in Figure 2.3.

As a validation of our results, we can note that our numerical solution matches with the analytical solution for the stationary state resulting from solving the convection-diffusion equation for this particular case. In particular, the analytical solution would give:

$$\phi(x) = \phi_{Left} + \frac{e^{Px/L_x} - 1}{e^P - 1}(\phi_{Right} - \phi_{Left})$$

This convergence can be seen in Figure 2.4, that illustrates the convergence to the stationary state (corresponding to the analytical solution of the convection-diffusion equation).

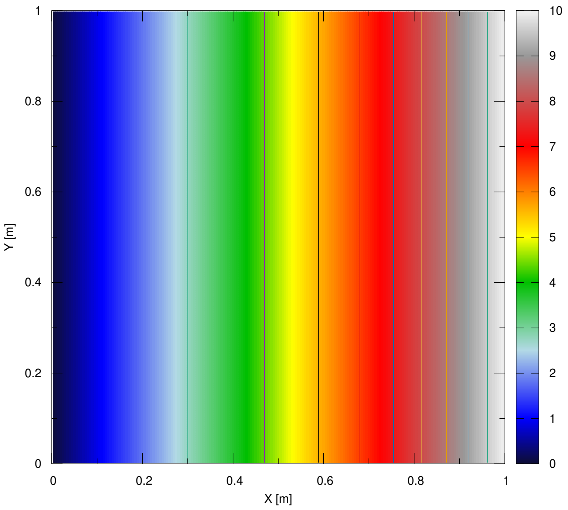


Figure 2.3: Distribution of ϕ throughout the material at the stationary state ($t = 272s$). Results obtained using $L_x = 1m$, $L_y = 1m$, $\rho = 2500kg/m^2$, $\Gamma = 10$, $\phi_{Initial} = 5$, $\phi_{Left} = 0$, $\phi_{Right} = 10$ and $U_0 = 0.01m/s$.

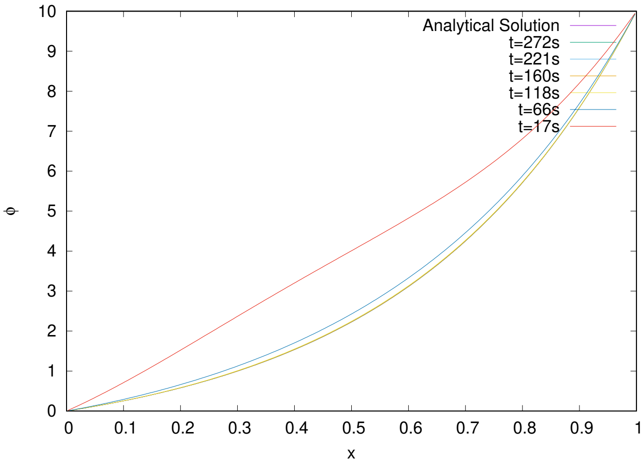
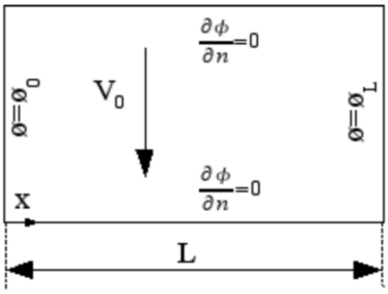


Figure 2.4: Distribution of $\phi(x, y = 0.5) = \phi(x)$ throughout the material at different instants of time. Results obtained using $L_x = 1m$, $L_y = 1m$, $\rho = 2500kg/m^2$, $\Gamma = 10$, $\phi_{Initial} = 5$, $\phi_{Left} = 0$, $\phi_{Right} = 10$ and $U_0 = 0.01m/s$.

2.2.2 Unidimensional flow with a unidimensional variation of the variable solved in the perpendicular direction of the flow

The second case is identical to the previous one except for the fact that the velocity of the field is vertical instead of being horizontal (as can be seen in the diagram of Figure 2.6). Thus, updating the value of the speed will result into the result of the simulation.



$\frac{\partial \phi}{\partial n}=0$
V_0
$\emptyset=\emptyset_0$
$\emptyset=\emptyset_1$
$\frac{\partial \phi}{\partial n}=0$
x
L

Figure 2.5: Schema of the boundary conditions for the case corresponding to the unidimensional flow with a unidimensional variation of the variable solved in the perpendicular direction of the flow.

Again, we have run the simulation until the stationary state was achieved (when the time derivative was smaller than 1.0e-6 at all the points of the plate). This has resulted into the following distribution of ϕ:

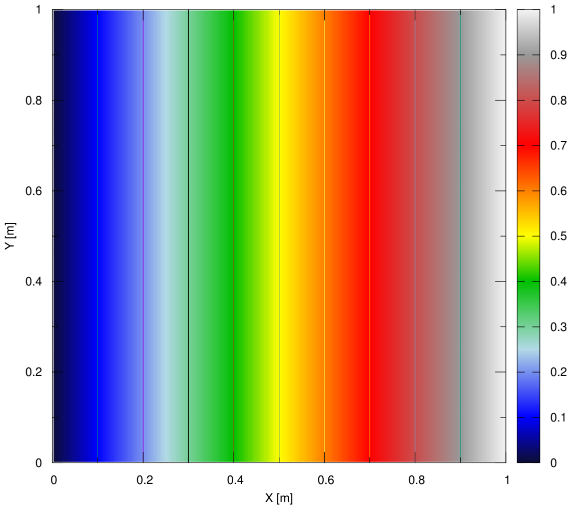


Figure 2.6: Distribution of ϕ throughout the material at the stationary state ($t = 91s$). Results obtained using $L_x = 1m$, $L_y = 1m$, $\rho = 2500kg/m^2$, $\Gamma = 10$, $\phi_{Initial} = 5$, $\phi_{Left} = 0$, $\phi_{Right} = 10$ and $U_0 = 0.01m/s$.

As in the previous case, the analytical solution for the stationary state resulting from solving the convection-diffusion equation in this situation can be obtained. In particular, the analytical solution would give:

$$\phi(x) = \phi_{Left} + \frac{\phi_{Right} - \phi_{Left}}{L_x} x$$

The following chart shows the quick convergence over time of the distribution of ϕ to the stationary solution:

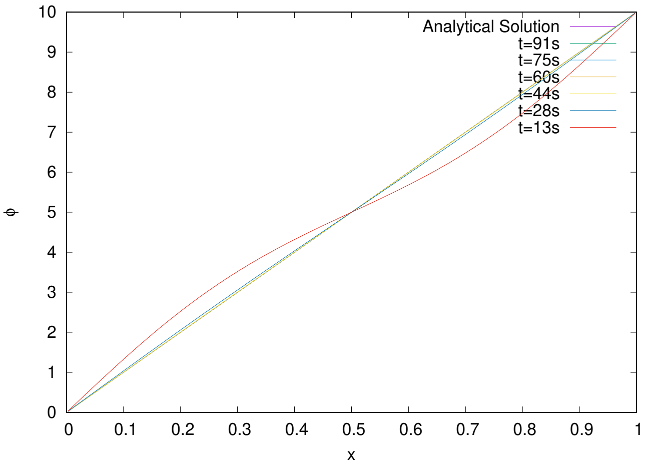


Figure 2.7: Distribution of $\phi(x, y = 0.5) = \phi(x)$ throughout the material at different instants of time. Results obtained using $L_x = 1m$, $L_y = 1m$, $\rho = 2500kg/m^2$, $\Gamma = 10$, $\phi_{Initial} = 5$, $\phi_{Left} = 0$, $\phi_{Right} = 10$ and $U_0 = 0.01m/s$.

2.2.3 Diagonal flow

In this case we still have the same rectangular material with an internal motion but with slightly different boundary conditions: there is a fixed value of ϕ at the top and left of the material, and another fixed value of ϕ at the bottom and right (see Section 2.1.1). A diagram sumarizing this case can be seen in the image below.

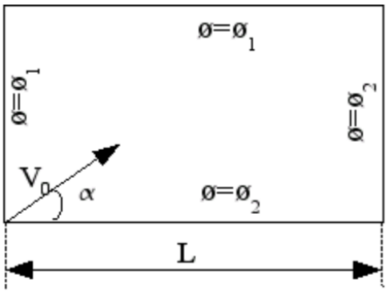
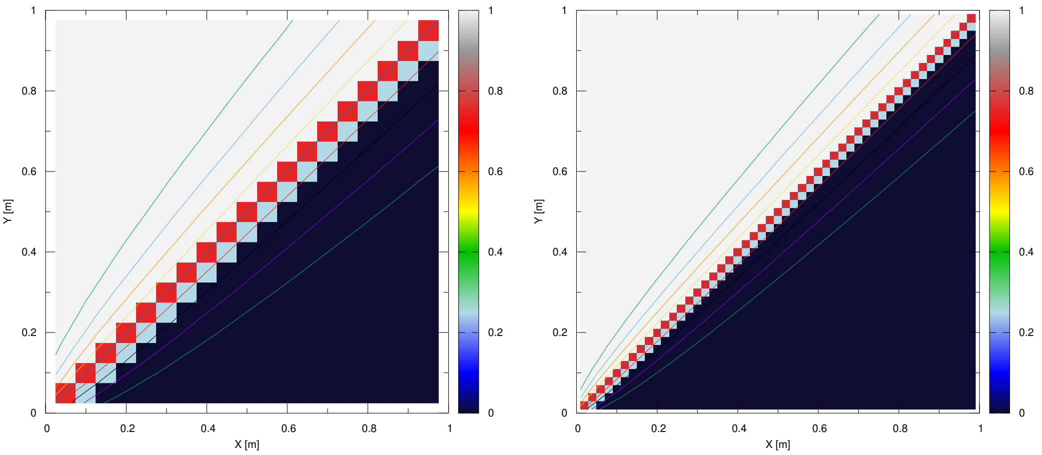


Figure 2.8: Schema of the boundary conditions for the case corresponding to the diagonal flow.

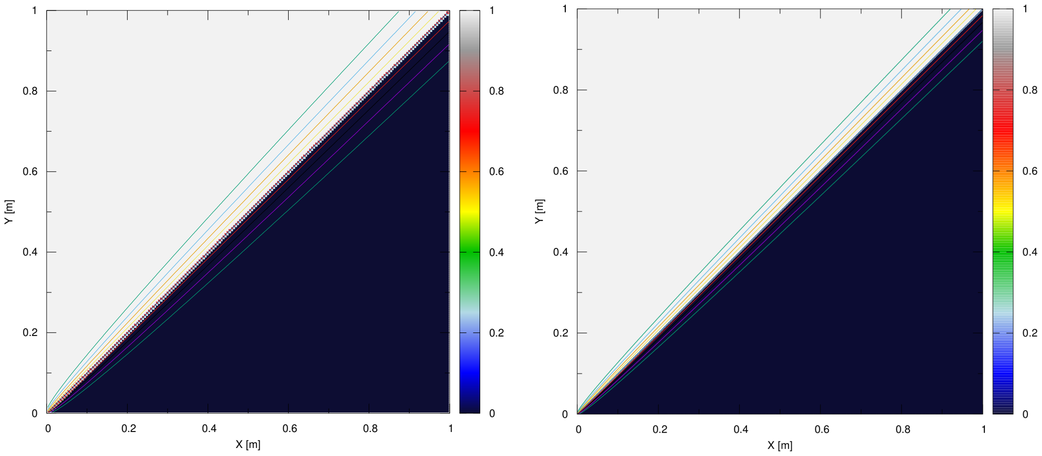
It is interesting to emphasize an important phenomenon that comes together with this case. We are talking about false diffusion, which is a consequence of the fact that the velocity is not parallel to the mesh that we have considered (it has a rotation equal to α degrees). Therefore, this results into an attenuation of the quick changes of the value of ϕ due to this false diffusion, which can be avoided by considering a finer mesh.

In order to illustrate this fact, we have run the simulation of this case considering a really big value of the Peclet number and an angle α equal to 45 degrees. Given that we are taking $L_x = L_y = 1$, our simulation should result into a sudden change of the value of ϕ, such that in the limit of $P \to \infty$, ϕ would be equal to ϕ_1 above the diagonal and ϕ_2 below (see the diagram of Figure 2.8). As a consequence of this sudden change, there would appear a false diffusion whose effects could be decreased by considering a finer mesh, i.e., bigger values of n_x and n_y. All these effects are clearly illustrated in Figure 2.8.



(a) $n_x = n_y = 20$

(b) $n_x = n_y = 50$



(c) $n_x = n_y = 200$

(d) $n_x = n_y = 500$

Figure 2.8: Distribution of ϕ throughout the material at the stationary state for different meshes. Results obtained using $L_x = 1m$, $L_y = 1m$, $P = 1.0\text{e}20$, $\phi_{Initial} = 0.5$, $\phi_1 = 0$ and $\phi_2 = 1$.

2.2.4 Solenoidal flow

Finally, the last case that we have simulated implies more complex boundary conditions that can be summarized, with the help of Figure 2.9, as follows:

$$\begin{cases} \phi_{Inlet}(x) = 1 + tanh\left[10(2x+1)\right], & \text{at the inlet} \\ \\ Q_{Outlet} = 0, & \text{at the outlet} \\ \\ \phi_{border} = 1 - tanh(10), & \text{at the top and lateral borders} \end{cases}$$

On the other side, the velocity field is defined as:

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

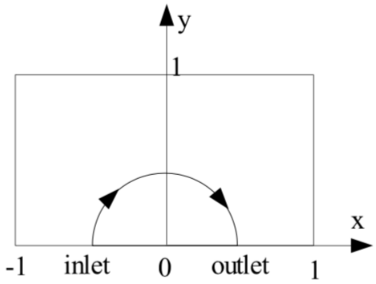
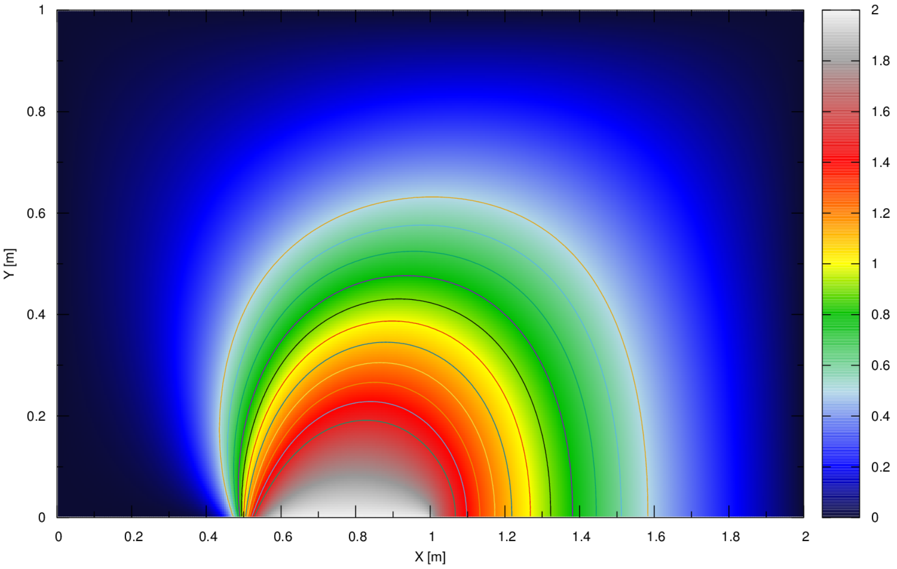
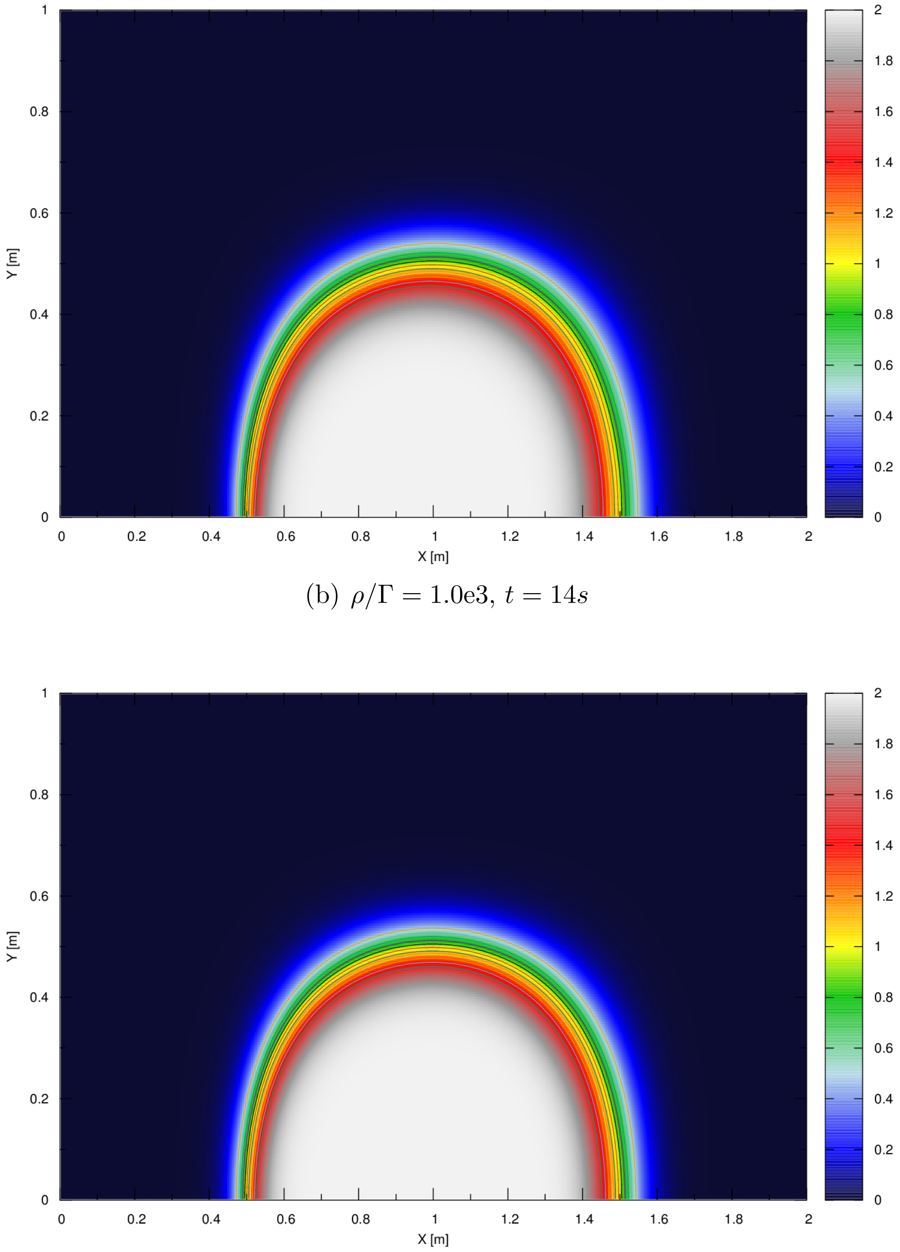


Figure 2.9: Schema of the boundary conditions for the case corresponding to the solenoidal flow.

Applying all these boundary conditions, we obtain the following distributions of ϕ at the stationary state for different values of ρ/Γ.



(a) $\rho/\Gamma = 10$, $t = 13s$



(b) $\rho/\Gamma = 1.0\text{e}3$, $t = 14s$

(c) $\rho/\Gamma = 1.0\text{e}6$, $t = 21s$

Figure 2.9: Distribution of ϕ throughout the material at the stationary state for different values of ρ/Γ. Results obtained using $L_x = 2m$, $L_y = 1m$ and $\phi_{Initial} = 0.5$.

It is interesting to note that when ρ/Γ becomes big, there appears a sudden change of the value of ϕ. This fact is clearly illustrated in Figure 2.10.

Let us note that, as happened with the diagonal flow corresponding to the previous case, now we are also dealing with a convective flux that is not parallel to the mesh considered. Thus, it becomes clear from Figure 2.10 that the bigger ρ/Γ is and the coarser the mesh is, the bigger impact false diffusion will have. This is why, in order to avoid this effect and show proper results in Figure 2.9, we have run the calculations using a fine mesh with $n_x = 400$ and $n_y = 200$.

It is interesting to study the impact that false diffusion would have on the results corresponding to $\rho/\Gamma = 1.0\text{e}6$ (that imply an abrupt change on the value of ϕ that

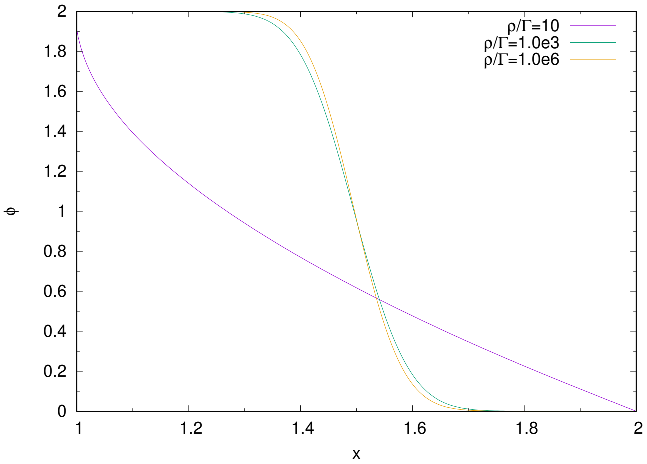
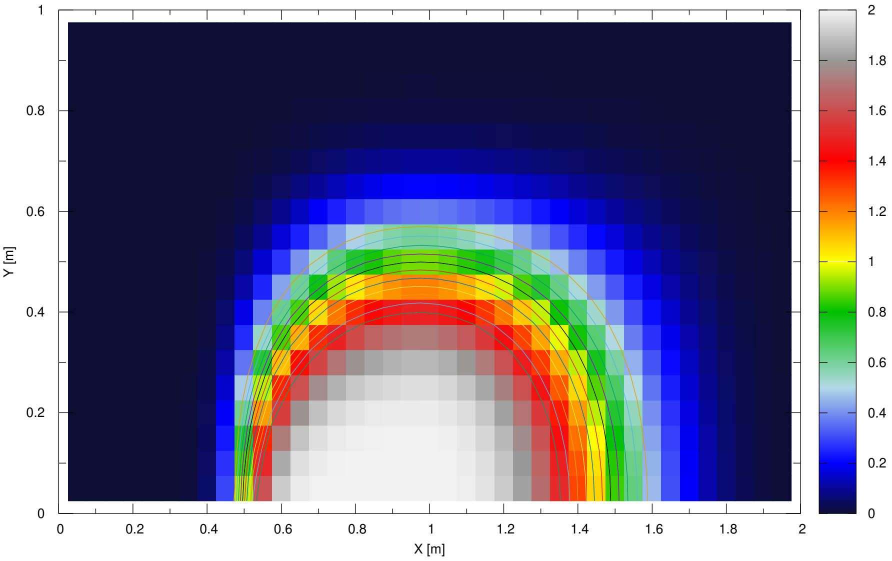
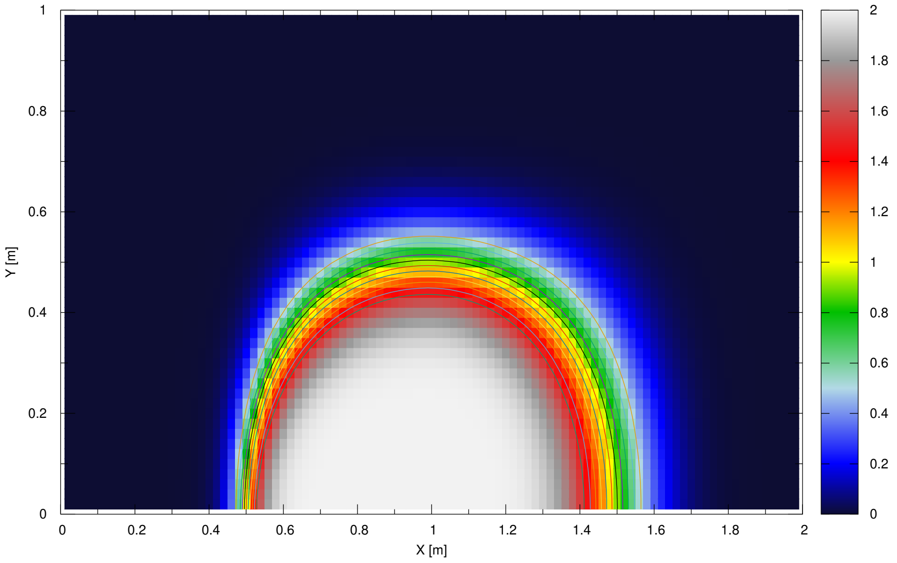


Figure 2.10: Distribution of ϕ throughout the outlet. Results obtained using $L_x = 2m$, $L_y = 1m$ and $\phi_{Initial} = 0.5$.

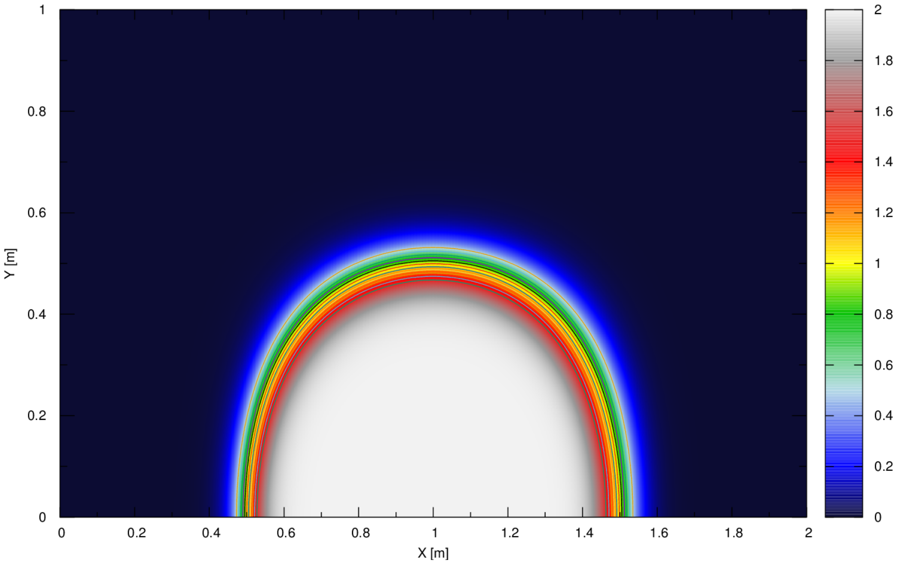
draws a semicircle going from the middle of the inlet to the middle of the outlet). With this aim, we have run exactly the same calculations but using different meshes. The distributions of ϕ that we have obtained can be found in 2.10.



(a) $n_x = 2n_y = 40$



(b) $n_x = 2n_y = 100$



(c) $n_x = 2n_y = 1000$

Figure 2.10: Distribution of ϕ throughout the material at the stationary state using different meshes. Results obtained using $\rho/\Gamma = 1.0e6$, $L_x = 2m$, $L_y = 1m$ and $\phi_{Initial} = 0.5$.

Comparing this images to its analogue of Figure 2.9 ($\rho/\Gamma = 1.0e6$), definitely justifies our choice of using a mesh with $n_x = 2n_y = 400$, given that it provides really exact results without consuming so much computing ressources. Although the impact of false diffusion related to the usage of coarse meshes is clearly shown in Figure 2.10, it can be put differently and made even more obvious. To do so, we have plotted in Figure 2.11 the variation of ϕ throughout the outlet calculated using different meshes. Hence, as a consequence of the false diffusion, which tends to smooth the sudden change that takes place around $x = 1.5m$, coarse meshes differ considerably with respect to the fine ones.

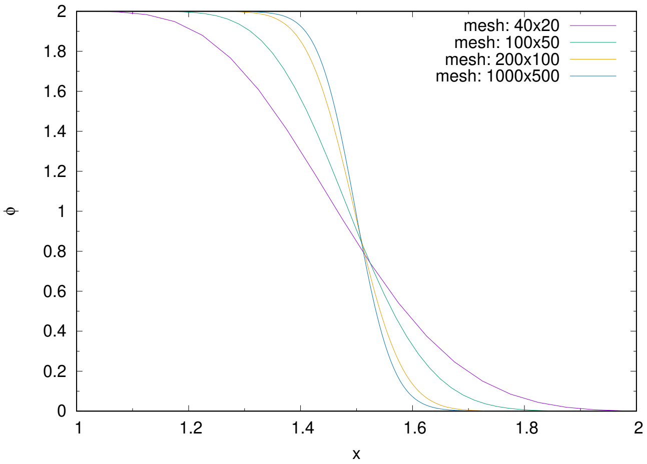


Figure 2.11: Distribution of ϕ at the outlet for different meshes corresponding to the stationary state. Results obtained using $\rho/\Gamma = 1.0\text{e}6$, $L_x = 2m$, $L_y = 1m$, $P = 1.0\text{e}20$ and $\phi_{Initial} = 0.5$.

Chapter 3

Lid-driven cavity

In this chapter we are going to simulate the flow of an incompressible fluid within a square cavity assuming that the velocity remains constant at all of the boundaries of the cavity. All these conditions are summarized in Figure 3.1.

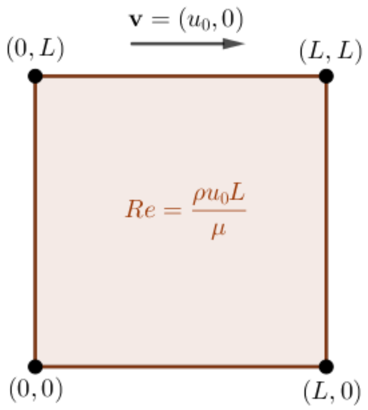


Figure 3.1: Schema of the plate amongst which the simulations are run.

In order to complete these task, we will have to solve the unsteady, incompressible Navier-Stokes equations in primitive variables. With that aim, we will introduce and apply the strategy followed in [4].

3.1 Discretization

3.1.1 Staggered meshes

In order to run this simulation, we will need to solve the incompressible Navier-Stokes equations. To do so, we will apply the well known fractional step method but, before detailing its formulation, we should make some comments regarding the meshes that are going to be used.

On one side, we will consider a regular square mesh as the one shown in Figure 0.1 and where we will calculate temperature and pressure but not the velocities. However, given that the gradient of the velocity at any point P_i would satisfy that:

$$\left(\frac{\partial v_x}{\partial x}, \frac{\partial v_y}{\partial y}\right) \simeq \left(\frac{v_{x_e} - v_{x_w}}{\Delta x}, \frac{v_{y_n} - v_{y_s}}{\Delta y}\right) \simeq \left(\frac{v_{x_E} - v_{x_W}}{\Delta x}, \frac{v_{y_N} - v_{y_S}}{\Delta y}\right),$$

then, any velocity field such that it alternates vertically and horizontally a pair of values, would result into an homogeneous gradient equal to zero throughout the plate, which, in turn, would not make any sense.

Therefore, in order to avoid this behaviour, we will use two more meshes (arising from the one shown in Figure 0.1). These two meshes will be called *staggered meshes*, and will be defined for the x and y-axis, respectively.

In particular, x-staggered mesh will have its points in the middle of all vertical walls of the points of the main mesh, whereas y-staggered mesh will have them in the middle of each horizontal wall. An example of this distribution is shown in Figure 3.2, where red and green-coloured points belong to x and y-staggered meshes, respectively.

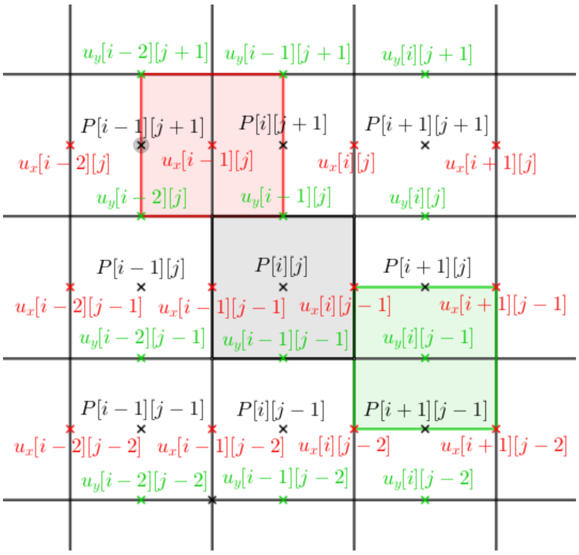


Figure 3.2: Diagram showing the staggered meshes corresponding to the x-axis (red) and y-axis (green) associated to the regular mesh that we are considering (see Figure 0.1). There are shown, as an example, three control volumes belonging to each of these meshes.

Hence, throughout this chapter axial velocities will be calculated in its corresponding staggered meshes and the rest of magnitudes in the standard one.

3.1.2 Fractional step method

In this section, we will summarize the fractional step method (FSM) and detail the calculations needed in its application. As we advanced, this method is going to be used to solve the unsteady and incompressible Navier-Stokes equations in primitive variables, which can be written as follows:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = \frac{1}{Re}\Delta \mathbf{v} - \nabla p, \tag{3.1}$$

where Re makes reference to the Reynolds number and is defined as:

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

Additionally, incompressibility results into the following version of the continuity equation:

$$\nabla \cdot \mathbf{v} = 0 \tag{3.2}$$

Hence, as seen in [4], FSM can be summarized into the following steps:

1. Evaluate the vector $\mathbf{R}(\mathbf{v}) := -(\mathbf{v} \cdot \nabla)\mathbf{v} + \frac{1}{Re}\Delta \mathbf{v}$
2. Evaluate the predictor velocity, $\mathbf{v}^p := \mathbf{v}^n + \Delta t \left(\frac{3}{2}\mathbf{R}(\mathbf{v}^n) - \frac{1}{2}\mathbf{R}(\mathbf{v}^{n-1})\right)$
3. Evaluate $\nabla \cdot \mathbf{v}^p$ and solve the equation of Poisson $\Delta \tilde{p} = \nabla \cdot \mathbf{v}^p$, in order to obtain the distribution of the field $\tilde{p}$
4. Obtain the velocity of the following instant of time as $\mathbf{v}^{n+1} = \mathbf{v}^p - \nabla \tilde{p}$

Let us explain in the following subsections how are we going to complete each of these steps by means of the FVM.

Evaluation of $\mathbf{R}(\mathbf{v})$

As we have already said, we will define a vector field $\mathbf{R}(\mathbf{v}) = (R_x(\mathbf{v}), R_y(\mathbf{v}))$ as follows:

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

or, equivalently,

$$\begin{cases} R_x(\mathbf{v}) = -(\mathbf{v} \cdot \nabla)v_x + \frac{1}{Re}\Delta v_x \\ R_y(\mathbf{v}) = -(\mathbf{v} \cdot \nabla)v_y + \frac{1}{Re}\Delta v_y \end{cases} \tag{3.3}$$

Bearing in mind that each of the components of the velocity is defined in a different staggered mesh (see Figure 3.2), then it becomes clear from Equation 3.3 that so will happen with the components of $\mathbf{R}(\mathbf{v})$. Hence, we will assume that the value of $R_x(\mathbf{v})$ at a certain point of the x-staggered mesh is equal to its integral across that particular control volume.

Thus, for the horizontal component of $\mathbf{R}(\mathbf{v})$ we have that

$$\begin{aligned} \int_V R_x(\mathbf{v})dV &= \int_V \left(-(\mathbf{v} \cdot \nabla)v_x + \frac{1}{Re}\Delta v_x\right) dV \\ &= -\int_V \nabla \cdot (\mathbf{v}v_x)dV + \frac{1}{Re}\int_V \nabla \cdot (\nabla v_x)dV \\ &= -\oint_{\partial V} v_x \mathbf{v} \cdot \hat{\mathbf{n}}dS + \frac{1}{Re}\oint_{\delta V} \nabla v_x \cdot \hat{\mathbf{n}}dS \end{aligned}$$

Furthermore, taking into account the structure of the mesh that we are considering (see Figure 3.2), we could make the following approximation:

$$\int_V R_x(\mathbf{v})dV \simeq \sum_{f\in F(\partial V)} \left(-v_{x_f} \left(v_{x_f}, v_{y_f} \right) \begin{pmatrix} n_{x_f} \\ n_{y_f} \end{pmatrix} S_f + \frac{1}{Re} \left(\left.\frac{\partial v_x}{\partial x}\right|_f , \left.\frac{\partial v_x}{\partial y}\right|_f \right) \begin{pmatrix} n_{x_f} \\ n_{y_f} \end{pmatrix} S_f \right)$$

and proceeding identically but with for the vertical component of $\mathbf{R}(\mathbf{v})$, we reach the following result:

$$\int_V R_y(\mathbf{v})dV \simeq \sum_{f\in F(\partial V)} \left(-v_{y_f} \left(v_{x_f}, v_{y_f} \right) \begin{pmatrix} n_{x_f} \\ n_{y_f} \end{pmatrix} S_f + \frac{1}{Re} \left(\left.\frac{\partial v_y}{\partial x}\right|_f , \left.\frac{\partial v_y}{\partial y}\right|_f \right) \begin{pmatrix} n_{x_f} \\ n_{y_f} \end{pmatrix} S_f \right)$$

Therefore, if we develop the summations across the northern, southern, eastern and western faces and approximate all derivatives, we can conclude that:

$$\begin{aligned} R_{x_P}(\mathbf{v}) = {} & -v^*_{x_n} v_{y_n} \Delta x + \frac{1}{Re}\frac{v_{x_N} - v_{x_P}}{\delta y_N}\Delta x + v^*_{x_s} v_{y_s} \Delta x + \frac{1}{Re}\frac{v_{x_S} - v_{x_P}}{\delta y_S}\Delta x \\ & -v^*_{x_e} v_{x_e} \Delta y + \frac{1}{Re}\frac{v_{x_E} - v_{x_P}}{\delta x_E}\Delta y + v^*_{x_w} v_{x_w} \Delta y + \frac{1}{Re}\frac{v_{x_W} - v_{x_P}}{\delta x_W}\Delta y \end{aligned}$$

and

$$\begin{aligned} R_{y_P}(\mathbf{v}) = {} & -v^*_{y_n} v_{y_n} \Delta x + \frac{1}{Re}\frac{v_{y_N} - v_{y_P}}{\delta y_N}\Delta x + v^*_{y_s} v_{y_s} \Delta x + \frac{1}{Re}\frac{v_{y_S} - v_{y_P}}{\delta y_S}\Delta x \\ & -v^*_{y_e} v_{x_e} \Delta y + \frac{1}{Re}\frac{v_{y_E} - v_{y_P}}{\delta x_E}\Delta y + v^*_{y_w} v_{x_w} \Delta y + \frac{1}{Re}\frac{v_{y_W} - v_{y_P}}{\delta x_W}\Delta y \end{aligned}$$

where v_{x_F} and v_{y_F} denote the velocity of the neighbour that is closest following the direction perpendicular to the face f, v_{x_f} and v_{y_f} denote the velocities at the corresponding faces of the control volume obtained by means of a linear interpolation, while $v^*_{x_f}$ and $v^*_{y_f}$ denote the same velocities but calculated using any schema one may chose, such as central difference, upwind, exponential... In our case, we have considered the upwind schema.

Evaluation of the predictor velocity, $\mathbf{v}^p$

The evaluation of the predictor velocity turns out to be rather simple. It is defined as:

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

or, equivalently,

$$\begin{cases} v_x^p = v_x^n + \Delta t \left(\frac{3}{2} R_x(\mathbf{v}^n) - \frac{1}{2} R_x(\mathbf{v}^{n-1}) \right) \\ v_y^p = v_y^n + \Delta t \left(\frac{3}{2} R_y(\mathbf{v}^n) - \frac{1}{2} R_y(\mathbf{v}^{n-1}) \right) \end{cases} \tag{3.4}$$

Therefore, when the vector field $\mathbf{R}(\mathbf{v}^n)$ is known at the previous and current instants of time, then the predictor velocity immediately follows.

Solution of the equation of Poisson for the pseudo-pressure

When the current predictor velocity is known, it is possible to calculate the pseudo-pressure, $\tilde{p} := p\Delta t$, by solving the following Posson equation:

$$\Delta\tilde{p} = \nabla \cdot \mathbf{v}^p \tag{3.5}$$

Thus, integrating it across any of the control volumes we have:

$$\int_V \nabla \cdot \nabla\tilde{p}dV = \int_V \nabla \cdot \mathbf{v}^p dV \tag{3.6}$$

and applying the divergence theorem,

$$\oint_{\partial V} \nabla\tilde{p} \cdot \hat{\mathbf{n}}dS = \oint_{\partial V} \mathbf{v}^p \cdot \hat{\mathbf{n}}dS \tag{3.7}$$

Finally, bearing in mind the structure of the regular mesh that we are considering, we can approximate Equation 3.7 to the following equality:

$$\sum_{f\in F(\partial V)} \left(\left.\frac{\partial\tilde{p}}{\partial x}\right|_f, \left.\frac{\partial\tilde{p}}{\partial y}\right|_f \right) \begin{pmatrix} n_{x_f} \\ n_{y_f} \end{pmatrix} S_f \simeq \sum_{f\in F(\partial V)} \left(v^p_{x_f}, v^p_{y_f} \right) \begin{pmatrix} n_{x_f} \\ n_{y_f} \end{pmatrix} S_f \tag{3.8}$$

Therefore,

$$\begin{aligned} \frac{\tilde{p}_N - \tilde{p}_P}{\delta y_N}\Delta x + \frac{\tilde{p}_S - \tilde{p}_P}{\delta y_S}\Delta x + &\frac{\tilde{p}_E - \tilde{p}_P}{\delta x_E}\Delta y + \frac{\tilde{p}_W - \tilde{p}_P}{\delta x_W}\Delta y = \\ &= v^p_{y_N}\Delta x - v^p_{y_S}\Delta x + v^p_{x_E}\Delta y - v^p_{x_W}\Delta y \end{aligned}$$

Thus, we can easily reached the final discretized equation to solve the equation of Poisson of the pseudo-pressure:

$$a_P\tilde{p} = a_N\tilde{p}_N + a_S\tilde{p}_S + a_E\tilde{p}_E + a_W\tilde{p}_W + b,$$

where

$$\begin{aligned} a_N &= \tfrac{\Delta x}{\delta y_N} \\ a_S &= \tfrac{\Delta x}{\delta y_S} \\ a_E &= \tfrac{\Delta y}{\delta x_E} \\ a_W &= \tfrac{\Delta y}{\delta x_W} \\ a_P &= a_N + a_S + a_E + a_W \\ b &= \Delta x(-v^p_{y_N} + v^p_{y_S}) + \Delta y(-v^p_{x_E}\Delta y + v^p_{x_W}\Delta y) \end{aligned}$$

Calculation of the velocity at the following time-step

This final step results quite simple, given that we just have to calculate the new components of the velocity (at its related staggered mesh) using the following expression:

$$\mathbf{v}^{n+1} = \mathbf{v}^p - \nabla\tilde{p} \tag{3.9}$$

or, equivalently,

$$\begin{cases} v_x^{n+1} = v_x^p - \frac{\partial\tilde{p}}{\partial x} \\ v_y^{n+1} = v_y^p - \frac{\partial\tilde{p}}{\partial y} \end{cases}$$

3.1.3 Boundary Conditions

Hence, we have just completed the review of the FSM detailing all of its calculations. Now, in order to run the simulation of the lid-driven cavity problem, we just need to implement this procedure in a C++ program conveniently applying all boundary conditions (velocities at the borders of the cavity). Let us now detail the application of these boundary conditions.

As presented in Figure 3.1, all boundary conditions that we are given are the velocities of the fluid at the borders of the cavity (see Table 3.1). On the other side, let us note that we are going to assume that the fluid is immobile throughout the cavity at the initial instant of time (this hypothesis does not have any impact on the results that we aim, given that we are only interested in the final stationary state).

Table 3.1: Summary of the boundary conditions at the borders of the square cavity.

Border	v_x	v_y
Top	v_0	0
Bottom	0	0
Right	0	0
Left	0	0

In order to illustrate the implementation of these boundary conditions into our algorithm, let us study the successive steps of the calculation separately.

Evaluation of R(v)

As we have already seen, we have that:

$$\begin{aligned}R_{x_P}(\mathbf{v}) = &-v^*_{x_n}v_{y_n}\Delta x + \frac{1}{Re}\frac{v_{x_N}-v_{x_P}}{\delta y_N}\Delta x + v^*_{x_s}v_{y_s}\Delta x + \frac{1}{Re}\frac{v_{x_S}-v_{x_P}}{\delta y_S}\Delta x \\ &-v^*_{x_e}v_{x_e}\Delta y + \frac{1}{Re}\frac{v_{x_E}-v_{x_P}}{\delta x_E}\Delta y + v^*_{x_w}v_{x_w}\Delta y + \frac{1}{Re}\frac{v_{x_W}-v_{x_P}}{\delta x_W}\Delta y\end{aligned} \tag{3.10}$$

and

$$\begin{aligned}R_{y_P}(\mathbf{v}) = &-v^*_{y_n}v_{y_n}\Delta x + \frac{1}{Re}\frac{v_{y_N}-v_{y_P}}{\delta y_N}\Delta x + v^*_{y_s}v_{y_s}\Delta x + \frac{1}{Re}\frac{v_{y_S}-v_{y_P}}{\delta y_S}\Delta x \\ &-v^*_{y_e}v_{x_e}\Delta y + \frac{1}{Re}\frac{v_{y_E}-v_{y_P}}{\delta x_E}\Delta y + v^*_{y_w}v_{x_w}\Delta y + \frac{1}{Re}\frac{v_{y_W}-v_{y_P}}{\delta x_W}\Delta y\end{aligned} \tag{3.11}$$

$R_x(\mathbf{v})$ will be clearly calculated at the x-staggered mesh and $R_y(\mathbf{v})$ at the y-staggered mesh. Therefore, it becomes clear that given a finite volume such that any of its faces lies on a wall of the cavity, then the velocity at that face will be equal to the corresponding value of Table 3.1. Figure 3.3 shows an example of this applied to the upper border of the cavity (the rest of borders are totally analogous).

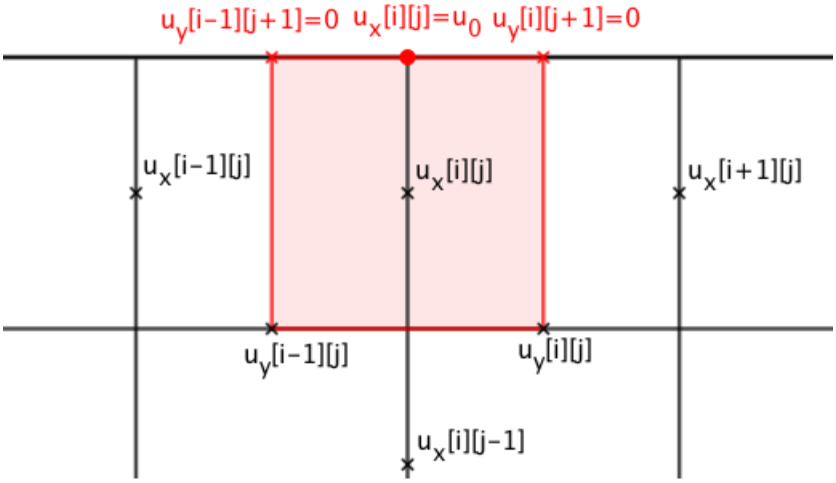


Figure 3.3: Schema of the boundary condition at the top border for $R_x(\mathbf{v})$.

Evaluation of the predictor velocity, $\mathbf{v}^p$

In Section 3.1.1, two different staggered meshes have been defined. Additionally each of the components of vectors $\mathbf{v}$, $\mathbf{R}(\mathbf{v})$ and $\mathbf{v}^p$ are calculated in its related staggered mesh. Thus, given the definition of these meshes and the boundary conditions summarized in Table 3.1, we will have that:

$$\begin{cases} v_x^p = 0, & \text{if the point lies at the right or left borders of the cavity,} \\ v_y^p = 0, & \text{if the point lies at the upper or lower borders of the cavity.} \end{cases}$$

Solution of the equation of Poisson for the pseudo-pressure

In the calculation of the pseudo-pressure, boundary conditions are applied to the coefficients of the discretized equations resulting from the Poisson equation. In particular, all coefficients related to a face that lies on a wall of the cavity, will be equal to zero. For example, if a point has its northern face lying on the upper border of the cavity, then $a_N = 0$.

Calculation of the velocity at the following time-step

Rather trivially, in the calculation of the velocity at the following time-step, we will just have to proceed exactly as we did with the predictor velocity, i.e., applying the following:

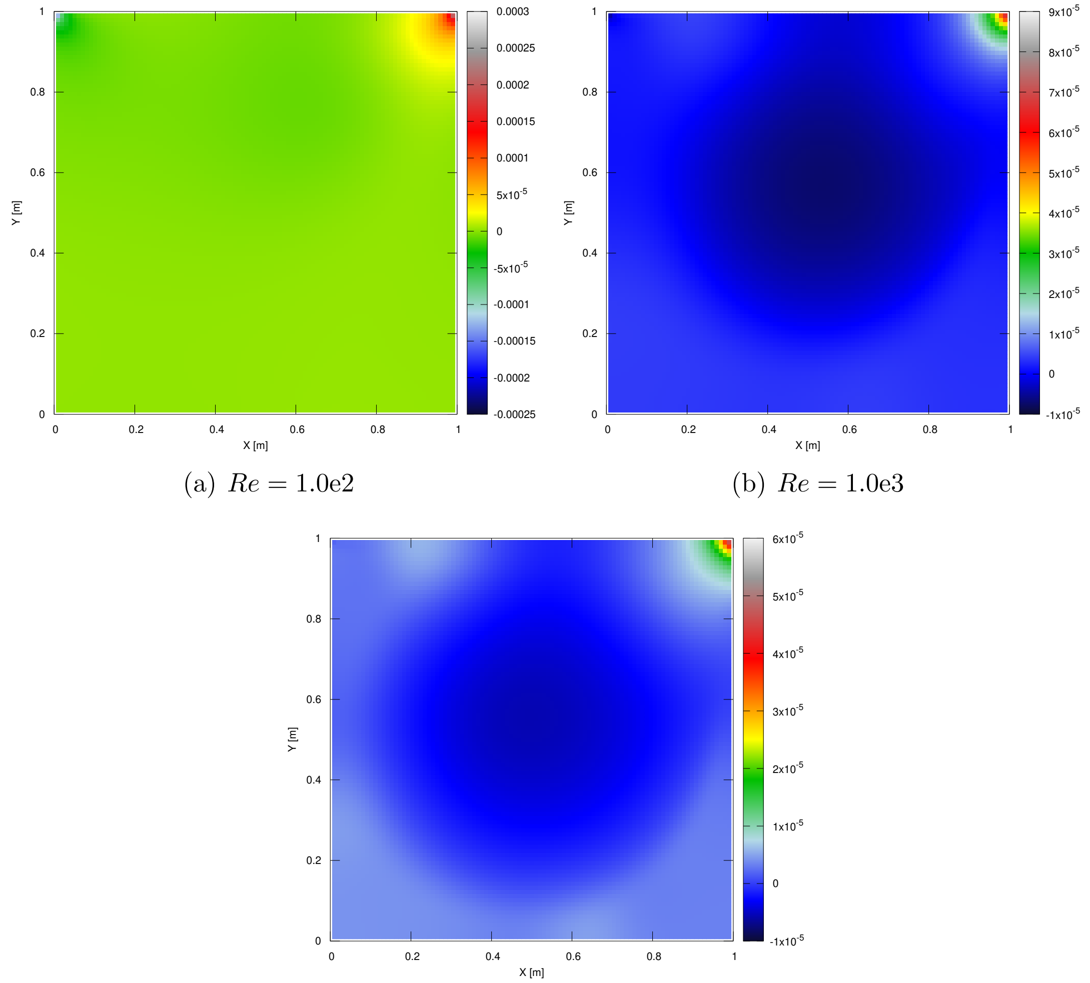
$$\begin{cases} v_x^{n+1} = 0, & \text{if the point lies at the right or left borders of the cavity,} \\ v_y^{n+1} = 0, & \text{if the point lies at the upper or lower borders of the cavity.} \end{cases}$$

3.2 Results

Finally, we have created a C++ program that runs all the calculations defined so far and adapts them to the lid-driven cavity problem arising from a given value of v_0 and Re.

In the images below, there can be found the pseudo-pressure corresponding to the stationary state of the lid-driven cavity problem (considering different boundary conditions and meshes). The criterion used to decide whether a certain system has reached the stationary state is:

$$\max_{P \in \text{Mesh}} \left\{ \frac{v_{xP}^{n+1} - v_{xP}^{n}}{\Delta t}, \frac{v_{yP}^{n+1} - v_{yP}^{n}}{\Delta t} \right\} \leq 1.0\text{e} - 6$$



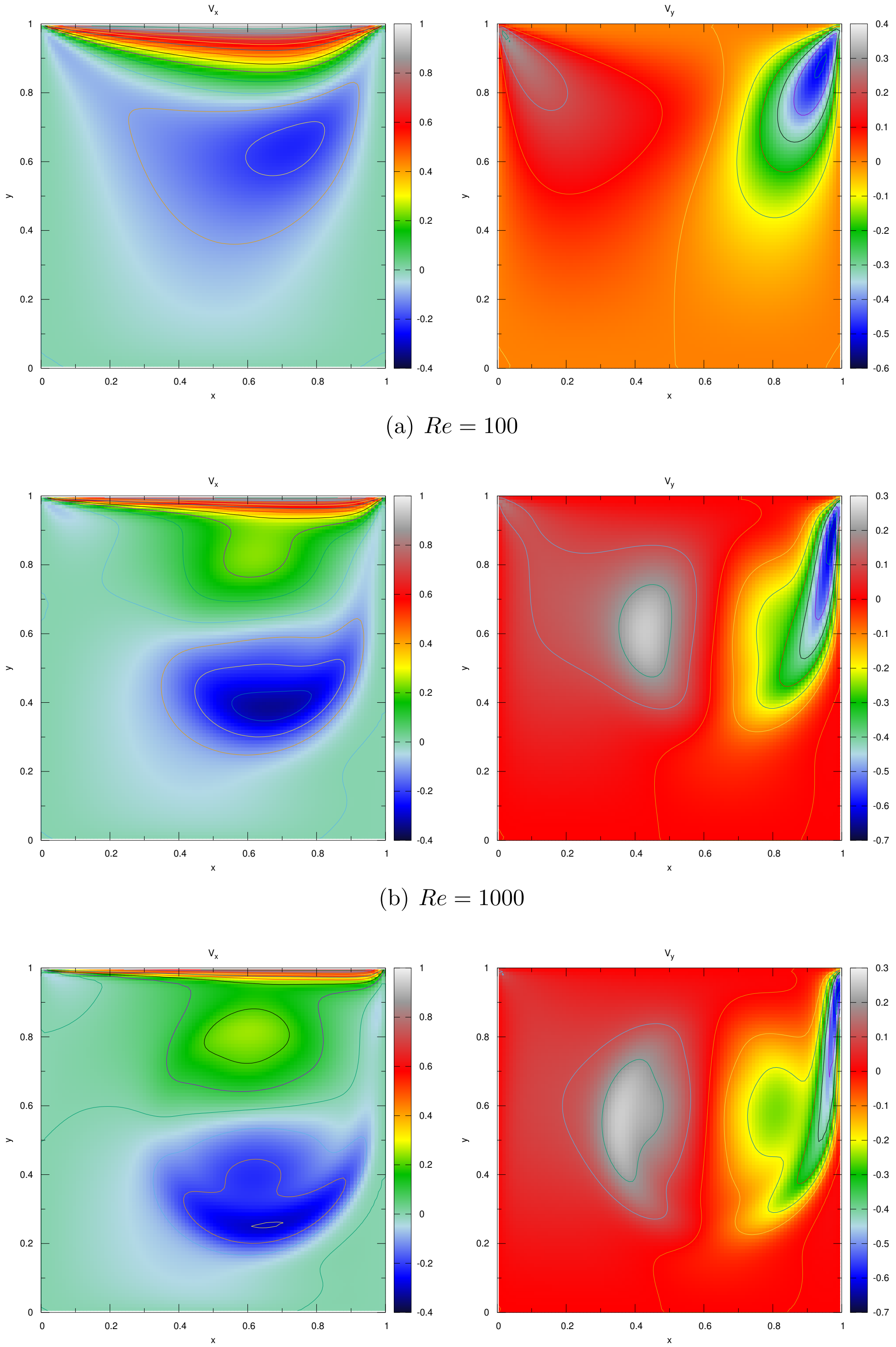
(a) $Re = 1.0\text{e}2$

(b) $Re = 1.0\text{e}3$

(c) $Re = 3.2\text{e}3$

Figure 3.4: Distribution of the pseudo-pressure, $\tilde{p}$, throughout the cavity at the stationary state for different values of the Reynolds number. Results obtained using a regular square mesh such that $n_x = n_y = 100$.

Additionally, the velocity fields corresponding to different values of Re, result into the following distributions:



(a) $Re = 100$

(b) $Re = 1000$

(c) $Re = 3200$

Figure 3.5: Distribution of the velocity throughout the material at the stationary state for different Reynolds number. Results obtained with $n_x = n_y = 100$ and $v_0 = \rho = \mu = L_x = L_y = 1.0$.

Chapter 4

Differentially heated cavity

In this chapter, we are going to simulate the natural convection of an incompressible fluid. More concretely, we will simulate the dynamics and heat transference of an incompressible two-dimensional fluid enclosed in a square cavity such as the one shown in Figure 4.1.

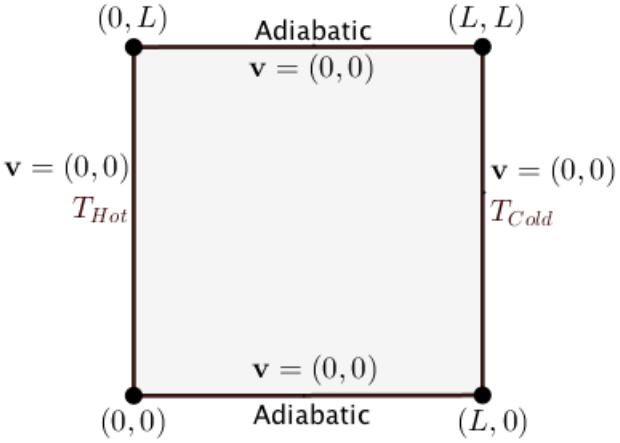


Figure 4.1: Diagram summarizing the boundary conditions corresponding to a differentially-heated cavity.

In order to complete this task, we will have to use numerical methods to solve the following set of partial differential equations:

Incompressible Continuity Equation. $\nabla \cdot \mathbf{v} = 0$

Momentum Equations. $\dfrac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = -\dfrac{1}{\rho_0}\nabla p + \dfrac{\mu_0}{\rho_0}\Delta \mathbf{v} + (1 - \beta_0(T - T_0))\mathbf{g}$

Energy Equation. $\dfrac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T = \alpha \Delta T$

Moreover, we are going to derive a slightly modified version of the FSM presented in Chapter 3, with the aim of implementing a C++ program that solves numerically a non-dimensional version of the continuity, momentum and energy equations. Additionally, the results obtained by our simulation will be validated with the bench mark provided by G. de Vahl Davis and I.P. Jones in [5, 6].

4.1 Discretization

Exactly as we did in Chapter 3, we will consider 3 meshes, the standard one shown in Figure 0.1 and two additional staggered meshes as the ones presented in Section 3.1.1.

As has been advanced, we will deal with a non-dimensional version of continuity, momentum and energy equations, in order to reach simpler and more general results. It is worth to mention that we are going to make use of the approximation of Boussinesq to simplify the effects of gravity on the fluid.

Obtaining the equivalent non-dimensional set of equations is as straightforward as applying to the governing equations the changes of variables shown in Table 4.1.

Table 4.1: Definition of non-dimensional variables..

Variable	**Non-dimensional equivalent**
Position	$\tilde{x} = \frac{x}{L}$, $\tilde{y} = \frac{y}{L}$
Velocity	$\tilde{v_x} = \frac{v_x}{v_0}$, $\tilde{v_y} = \frac{v_y}{v_0}$, where $v_0 = {}^{\alpha}/_{L}$
Time	$\tilde{t} = \frac{t}{{}^{L}/_{v_0}}$
Temperature	$\tilde{T} = \frac{T - T_{Cold}}{T_{Hot} - T_{Cold}}$
Pressure	$\tilde{p} = \frac{p}{\rho v_0^2}$

After doing so, the following equations are reached:

$$\nabla \cdot \tilde{\mathbf{v}} = 0 \tag{4.1}$$

$$\frac{\partial \tilde{\mathbf{v}}}{\partial \tilde{t}} + (\tilde{\mathbf{v}} \cdot \tilde{\nabla})\tilde{\mathbf{v}} = -\tilde{\nabla}\tilde{p} + Pr\tilde{\Delta}\tilde{\mathbf{v}} + RaPr\tilde{T}\mathbf{z} \tag{4.2}$$

$$\frac{\partial \tilde{T}}{\partial \tilde{t}} + \tilde{\mathbf{v}} \cdot \tilde{\nabla}\tilde{T} = \tilde{\Delta}\tilde{T} \tag{4.3}$$

where Prandtl number, $Pr = \frac{C_p \mu}{k}$, and Rayleigh number, $Ra = \frac{g\beta\Delta T L^3}{\nu\alpha}$, are used.

Hence, we are finally ready to describe the numerical methods applied to solve equations 4.1, 4.2 and 4.3. In fact, energy and momentum equations are going to be solved separately. On one side, energy equation will be treated exactly as proposed in [1]. However, momentum equation will be solved using a slightly modified version of the FSM presented in [4]. Thus, we are only going to detail the derivation of this FSM applied to the non-dimensional momentum equation arising from a two-dimensional flow under the Boussinesq approximation.

In order to avoid using an unnecessarily overcomplicated notation, from now and until the end of this chapter, we are going to identify non-dimensional variables exactly as its primitive analogues, i.e., we are going to avoid all tildes and systematically refer to non-dimensional variables (even if it is not explicitly noted).

4.1.1 Fractional step method

FSM is based on the Helmholtz-Hodge decomposition theorem, which will be used to uniquely project vectors into two orthogonal subspaces: a divergence-free one and another corresponding to pure gradient fields.

Let us define a linear operator Π that projects vector fields onto divergence-free subspaces or, equivalently, such that given any vector field $\mathbf{a}$, then $\nabla \cdot \Pi(\mathbf{a}) = 0$. Throughout this section, two properties of Π are going to be used:

Proposition 1. *If a vector field* $\mathbf{a}$ *satisfies that* $\nabla \cdot \mathbf{a} = 0$*, then* $\Pi(\mathbf{a}) = \mathbf{a}$.

Proposition 2. *Given any scalar field* ϕ*, then* $\Pi(\nabla\phi) = 0$.

Hence, two immediate consequences arising from these properties can be stated. On the one side, we can assert from Proposition 1 that:

$$\Pi(\nabla p) = 0.$$

On the other side, bearing in mind the incompressibility of the flow and recalling Proposition 2, we can affirm that:

$$\Pi\left(\frac{\partial \mathbf{v}}{\partial t}\right) = \frac{\partial \mathbf{v}}{\partial t}. \tag{4.4}$$

Thus, linearity of Π allows us to conclude that:

$$\Pi\left(\frac{\partial \mathbf{v}}{\partial t} + \nabla p\right) = \frac{\partial \mathbf{v}}{\partial t}. \tag{4.5}$$

Additionally, applying Π to both sides of Equation 4.2 and making use of equations 4.4 and 4.5 leads us to:

$$\frac{\partial \mathbf{v}}{\partial t} = \Pi\left(-(\mathbf{v}\cdot\nabla)\mathbf{v} + Pr\Delta\mathbf{v} + RaPrT\mathbf{z}\right). \tag{4.6}$$

Finally, Equation 4.6 can be substituted into Equation 4.2 as follows:

$$\frac{\partial \mathbf{v}}{\partial t} = -\nabla p + \mathbf{R}(\mathbf{v}), \tag{4.7}$$

where

$$\mathbf{R}(\mathbf{v}) := -(\mathbf{v}\cdot\nabla)\mathbf{v} + Pr\Delta\mathbf{v} + RaPrT\mathbf{z}. \tag{4.8}$$

In spite of the fact that compared to the FSM presented in Chapter 3 we are using a slightly different expression for $\mathbf{R}(\mathbf{v})$ (see Equation 4.8), the remaining mathematical deduction of the method totally coincides with the one presented before. Thus, we are not going to rewrite its details, but will just summarize its application in the following subsection.

Hence, as we have just seen, FSM can be summarized into the following steps:

1. Evaluate the vector $\mathbf{R}(\mathbf{v}) := -(\mathbf{v}\cdot\nabla)\mathbf{v} + Pr\Delta\mathbf{v} + RaPrT\mathbf{z}$
2. Evaluate the predictor velocity, $\mathbf{v}^p := \mathbf{v}^n + \Delta t\left(\frac{3}{2}\mathbf{R}(\mathbf{v}^n) - \frac{1}{2}\mathbf{R}(\mathbf{v}^{n-1})\right)$
3. Evaluate $\nabla\cdot\mathbf{v}^p$ and solve the equation of Poisson $\Delta\bar{p} = \nabla\cdot\mathbf{v}^p$, in order to obtain the distribution of the field $\bar{p}$
4. Obtain the velocity of the following instant of time as $\mathbf{v}^{n+1} = \mathbf{v}^p - \nabla\bar{p}$

Let us explain in the following sections how are we going to complete each of these steps by means of the FVM.

Evaluation of $\mathbf{R}(\mathbf{v})$

As we have already said, we will define a vector field $\mathbf{R}(\mathbf{v}) = (R_x(\mathbf{v}), R_y(\mathbf{v}))$ as follows:

$$\mathbf{R}(\mathbf{v}) := -(\mathbf{v} \cdot \nabla)\mathbf{v} + Pr\Delta\mathbf{v} + RaPrT\mathbf{z}$$

or, equivalently,

$$\begin{cases} R_x(\mathbf{v}) = -(\mathbf{v} \cdot \nabla)v_x + Pr\Delta v_x \\ R_y(\mathbf{v}) = -(\mathbf{v} \cdot \nabla)v_y + Pr\Delta v_y + RaPrT \end{cases} \tag{4.9}$$

Bearing in mind that each of the components of the velocity is defined in a different staggered mesh (see Figure 3.2), then it becomes clear from Equation 4.9 that so will happen with the components of $\mathbf{R}(\mathbf{v})$. Hence, we will assume that the value of $R_x(\mathbf{v})$ at a certain point of the x-staggered mesh is equal to its integral across that particular control volume.

Thus, for the vertical component of $\mathbf{R}(\mathbf{v})$ we have that

$$\begin{aligned} \int_V R_y(\mathbf{v})dV &= \int_V \left(-(\mathbf{v} \cdot \nabla)v_y + Pr\Delta v_y + RaPrT\right) dV \\ &= -\int_V \nabla \cdot (\mathbf{v}v_y)dV + Pr\int_V \nabla \cdot (\nabla v_y)dV + RaPrT\int_V 1 \cdot dV \\ &= -\oint_{\partial V} v_y\mathbf{v} \cdot \mathbf{n}dS + Pr\oint_{\delta V} \nabla v_y \cdot \mathbf{n}dS + RaPrTV \end{aligned}$$

Furthermore, taking into account the structure of the mesh that we are considering (see Figure 3.2), we could make the following approximation:

$$\begin{aligned} \int_V R_y(\mathbf{v})dV \simeq &\sum_{f\in F(\partial V)} \left(-v_{y_f}\left(v_{x_f}, v_{y_f}\right)\begin{pmatrix} n_{x_f} \\ n_{y_f}\end{pmatrix} S_f + Pr\left(\left.\frac{\partial v_y}{\partial x}\right|_f, \left.\frac{\partial v_y}{\partial y}\right|_f\right)\begin{pmatrix} n_{x_f} \\ n_{y_f}\end{pmatrix} S_f\right) \\ &+ RaPrT\Delta x\Delta y \end{aligned}$$

and proceeding identically but with for the horizontal component of $\mathbf{R}(\mathbf{v})$, we reach the following result:

$$\int_V R_x(\mathbf{v})dV \simeq \sum_{f\in F(\partial V)} \left(-v_{x_f}\left(v_{x_f}, v_{y_f}\right)\begin{pmatrix} n_{x_f} \\ n_{y_f}\end{pmatrix} S_f + Pr\left(\left.\frac{\partial v_x}{\partial x}\right|_f, \left.\frac{\partial v_x}{\partial y}\right|_f\right)\begin{pmatrix} n_{x_f} \\ n_{y_f}\end{pmatrix} S_f\right)$$

Therefore, if we develop the summations across the northern, southern, eastern and western faces and approximate all derivatives, we can conclude that:

$$\begin{aligned} R_{x_P}(\mathbf{v}) = &-v^*_{x_n}v_{y_n}\Delta x + Pr\frac{v_{x_N} - v_{x_P}}{\delta y_N}\Delta x + v^*_{x_s}v_{y_s}\Delta x + Pr\frac{v_{x_S} - v_{x_P}}{\delta y_S}\Delta x \\ &-v^*_{x_e}v_{x_e}\Delta y + Pr\frac{v_{x_E} - v_{x_P}}{\delta x_E}\Delta y + v^*_{x_w}v_{x_w}\Delta y + Pr\frac{v_{x_W} - v_{x_P}}{\delta x_W}\Delta y \end{aligned}$$

and

$$\begin{aligned} R_{y_P}(\mathbf{v}) = &-v^*_{y_n}v_{y_n}\Delta x + Pr\frac{v_{y_N} - v_{y_P}}{\delta y_N}\Delta x + v^*_{y_s}v_{y_s}\Delta x + Pr\frac{v_{y_S} - v_{y_P}}{\delta y_S}\Delta x \\ &-v^*_{y_e}v_{x_e}\Delta y + Pr\frac{v_{y_E} - v_{y_P}}{\delta x_E}\Delta y + v^*_{y_w}v_{x_w}\Delta y + Pr\frac{v_{y_W} - v_{y_P}}{\delta x_W}\Delta y \\ &+RaPrT\Delta x\Delta y \end{aligned}$$

where v_{x_F} and v_{y_F} denote the velocity of the neighbour that is closest following the direction perpendicular to the face f, v_{x_f} and v_{y_f} denote the velocities at the corresponding faces of the control volume obtained by means of a linear interpolation, while $v^*_{x_f}$ and $v^*_{y_f}$ denote the same velocities but calculated using any schema one may chose, such as central difference, upwind, exponential... In our case, we have considered the upwind schema.

Evaluation of the predictor velocity, $\mathbf{v}^p$

The evaluation of the predictor velocity turns out to be rather simple. It is defined as:

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

or, equivalently,

$$\begin{cases} v_x^p = v_x^n + \Delta t \left(\frac{3}{2} R_x(\mathbf{v}^n) - \frac{1}{2} R_x(\mathbf{v}^{n-1}) \right) \\ v_y^p = v_y^n + \Delta t \left(\frac{3}{2} R_y(\mathbf{v}^n) - \frac{1}{2} R_y(\mathbf{v}^{n-1}) \right) \end{cases} \tag{4.10}$$

Therefore, when the vector field $\mathbf{R}(\mathbf{v}^n)$ is known at the previous and current instants of time, then the predictor velocity immediately follows.

Solution of the equation of Poisson for the pseudo-pressure

When the current predictor velocity is known, it is possible to calculate the pseudo-pressure, $\bar{p} := p\Delta t$, by solving the following Posson equation:

$$\Delta \bar{p} = \nabla \cdot \mathbf{v}^p$$

Thus, integrating it across any of the control volumes we have:

$$\int_V \nabla \cdot \nabla \bar{p} dV = \int_V \nabla \cdot \mathbf{v}^p dV \tag{4.11}$$

and applying the divergence theorem,

$$\oint_{\partial V} \nabla \bar{p} \cdot \mathbf{n} dS = \oint_{\partial V} \mathbf{v}^p \cdot \mathbf{n} dS \tag{4.12}$$

Finally, bearing in mind the structure of the regular mesh that we are considering, we can approximate Equation 4.12 to the following equality:

$$\sum_{f \in F(\partial V)} \left(\left.\frac{\partial \bar{p}}{\partial x}\right|_f, \left.\frac{\partial \bar{p}}{\partial y}\right|_f \right) \begin{pmatrix} n_{x_f} \\ n_{y_f} \end{pmatrix} S_f \simeq \sum_{f \in F(\partial V)} \left(v^p_{x_f}, v^p_{y_f} \right) \begin{pmatrix} n_{x_f} \\ n_{y_f} \end{pmatrix} S_f \tag{4.13}$$

Therefore,

$$\frac{\bar{p}_N - \bar{p}_P}{\delta y_N} \Delta x + \frac{\bar{p}_S - \bar{p}_P}{\delta y_S} \Delta x + \frac{\bar{p}_E - \bar{p}_P}{\delta x_E} \Delta y + \frac{\bar{p}_W - \bar{p}_P}{\delta x_W} \Delta y = v^p_{y_N} \Delta x - v^p_{y_S} \Delta x + v^p_{x_E} \Delta y - v^p_{x_W} \Delta y$$

Thus, we can easily reached the final discretized equation to solve the equation of Poisson of the pseudo-pressure:

$$a_P \bar{p} = a_N \bar{p}_N + a_S \bar{p}_S + a_E \bar{p}_E + a_W \bar{p}_W + b,$$

where

$$\begin{aligned} a_N &= {}^{\Delta x}\!/_{\delta y_N} \\ a_S &= {}^{\Delta x}\!/_{\delta y_S} \\ a_E &= {}^{\Delta y}\!/_{\delta x_E} \\ a_W &= {}^{\Delta y}\!/_{\delta x_W} \\ a_P &= a_N + a_S + a_E + a_W \\ b &= \Delta x(-v^p_{y_N} + v^p_{y_S}) + \Delta y(-v^p_{x_E} y + v^p_{x_W}) \end{aligned}$$

Calculation of the velocity at the following time-step

This final step results quite simple, given that we just have to calculate the new components of the velocity (at its related staggered mesh) using the following expression:

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

or, equivalently,

$$\begin{cases} u_x^{n+1} = v_x^p - {}^{\partial \bar{p}}\!/_{\partial x} \\ u_y^{n+1} = v_y^p - {}^{\partial \bar{p}}\!/_{\partial y} \end{cases}$$

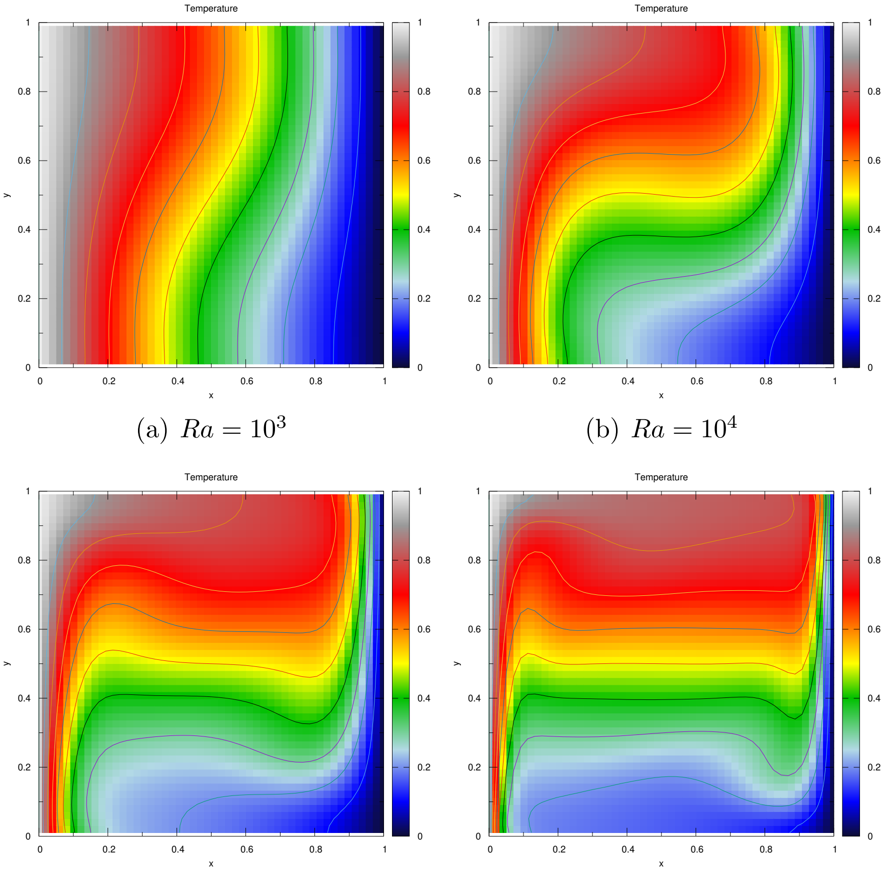
Hence, we have just completed the review of the FSM detailing all of its calculations. Now, in order to run the simulation of the differentially heated cavity problem, we just need to implement this procedure in a C++ program conveniently applying all boundary conditions (velocities and temperatures at the borders of the cavity). The application of these boundary conditions will be studied in the following section.

4.2 Results

Finally, we have created a C++ program that runs all the calculations defined so far and adapts them to the differentially heated problem arising from a given value of Pr and Ra.

In the images below, there can be found the distributions of temperature and velocities corresponding to the stationary states of several differentially heated cavities obtained using $Pr = 0.71$ (air) and $Ra \in \{10^3, 10^4, 10^5, 10^6\}$. The criterion used to decide whether a certain system has reached the stationary state is:

$$\max_{P \in \text{Mesh}} \left\{ \frac{u_{xP}^{n+1} - u_{xP}^{n}}{\Delta t}, \frac{u_{yP}^{n+1} - u_{yP}^{n}}{\Delta t}, \frac{T_P^{n+1} - T_P^{n}}{\Delta t} \right\} \leq 1.0\mathrm{e} - 6$$



(a) $Ra = 10^3$

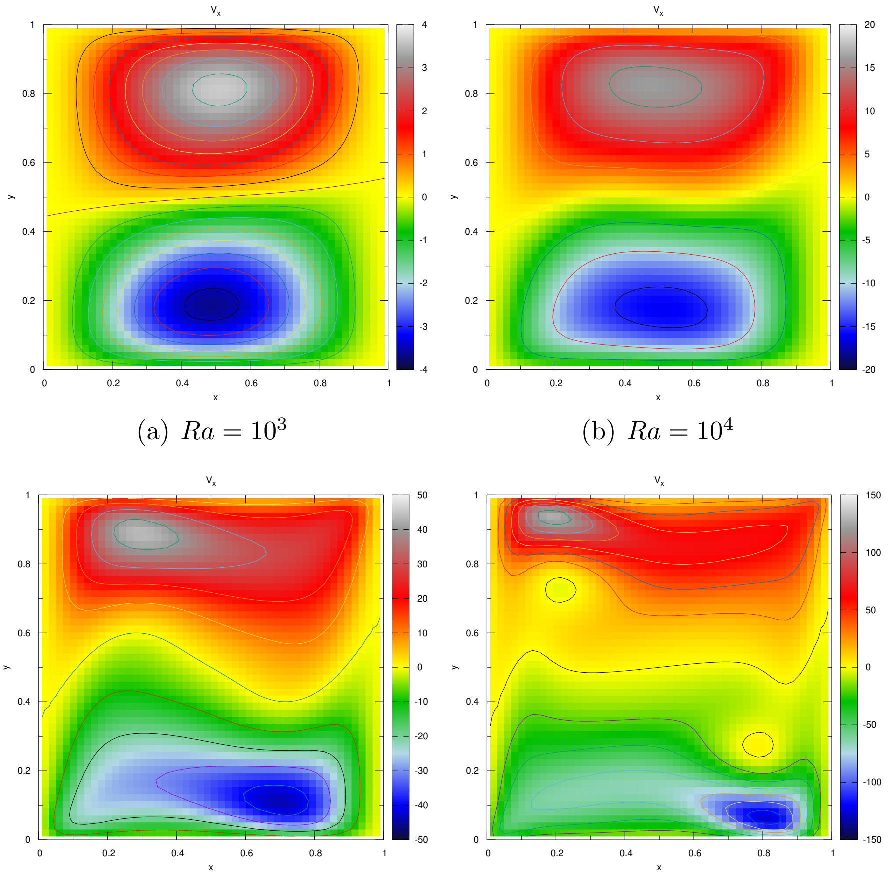
(b) $Ra = 10^4$

(c) $Ra = 10^5$

(d) $Ra = 10^6$

Figure 4.2: Distribution of the non-dimensional temperature throughout the cavity for different values of the Rayleigh number. Results obtained with $Pr = 0.71$ and $n_x = n_y = 51$.

Additionally, the velocity fields corresponding to different meshes (either finer or coarser) and different values of Re, result into the following distributions:



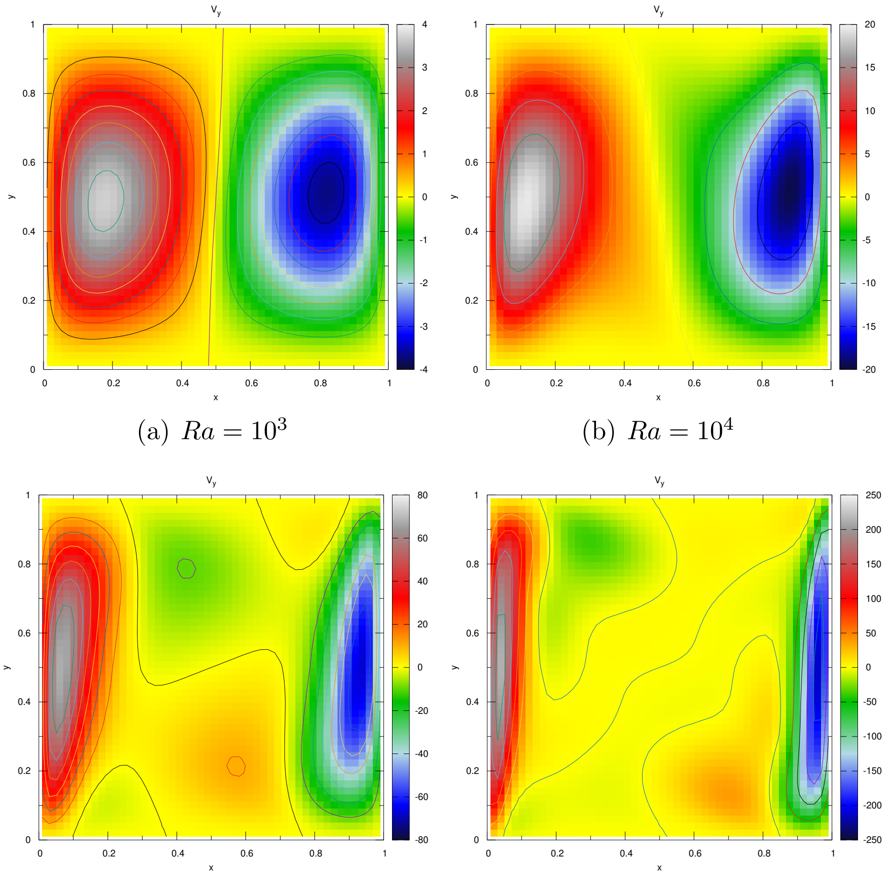
(a) $Ra = 10^3$

(b) $Ra = 10^4$

(c) $Ra = 10^5$

(d) $Ra = 10^6$

Figure 4.3: Distribution of the non-dimensional horizontal velocity throughout the cavity for different values of the Rayleigh number. Results obtained with $Pr = 0.71$ and $n_x = n_y = 51$.



(a) $Ra = 10^3$

(b) $Ra = 10^4$

(c) $Ra = 10^5$

(d) $Ra = 10^6$

Figure 4.4: Distribution of the non-dimensional vertical velocity throughout the cavity for different values of the Rayleigh number. Results obtained with $Pr = 0.71$ and $n_x = n_y = 51$.

Chapter 5

Burgers equation

Given the extremely high computational cost of directly numerically simulate turbulence resolving Navier-Stokes equation, there has been proposed a simplified model that shares many of its properties. Thus, resolving it may be an interesting first step to tackle turbulence. We are talking about Burgers equation, which will be introduced, discretized and resolved by means of spectral and pseudo-spectral methods.

Hence, Burgers equation can be naturally obtained as a simplification If we consider the non-dimensional Navier-Stokes equation,

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = \frac{1}{Re}\Delta \mathbf{v} - \nabla p, \text{ where } Re = \frac{\rho V_0 L}{\mu}, \tag{5.1}$$

we can easily reach Burgers equation substituting the gradient of pressure by a forcing term, $\mathbf{f}$, as follows:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = \frac{1}{Re}\Delta \mathbf{v} + \mathbf{f} \tag{5.2}$$

The aim of this chapter is to present the ressolution of the one and two-dimensional Burgers equation. However, differently to what we did in the preceding cases, we are going to make use of spectral and pseudo-spectral numerical methods, which are based on the utilization of Fourier transforms in order to optimize the calculations.

When it comes to the derivation of the spectral methods, our work will be based on the results obtained in [7]. However, in the implementation of pseudo-spectral methods we have deducted its expression and implemented it by means of the open-source library FFTW (see [9]), which calculates Fourier transforms using a highly efficient algorithm. Additionally, de-aliasing techniques have been implemented for the pseudo-spectral methods taking care of the remarks made by F. Lu in [8].

5.1 Discretization

In order to simplify the presentation of the discretization applied to the Burgers equation, we are just going to detail the one-dimensional case and, once it has been finished, make some comments relative to the general one.

Hence, we will work with the following PDE to simulate a one-dimensional "turbulent" motion:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2} + f \tag{5.3}$$

Following the spectral method presented in [7], we will start applying a discrete Fourier decomposition to the velocity function, leading to the following approximation:

$$u(x,t) = \sum_{-\infty \leq k \leq \infty} \hat{u}_k(t)e^{ikx} \simeq \sum_{k=-N}^{k=N} \hat{u}_k(t)e^{ikx}, \text{ where } \hat{u}_k(t) \in \mathbb{C} \tag{5.4}$$

Obviously, the greater N is considered, the better solution (implying a higher computational cost) is achieved. From the approximation of Equation 5.4, it results straightforward to reach a version of the Burgers equation in the Fourier space by rewriting Equation 5.3 as:

$$\sum_{k=-N}^{k=N} \frac{d\hat{u}_k}{dt}e^{ikx} + \left(\sum_{k=-N}^{k=N} \hat{u}_k e^{ikx}\right) \cdot \left(\sum_{k=-N}^{k=N} ik\hat{u}_k e^{ikx}\right) = \frac{-1}{Re}\sum_{k=-N}^{k=N} k^2\hat{u}_k e^{ikx} + \sum_{k=-N}^{k=N} F_k e^{ikx} \tag{5.5}$$

or, alternatively,

$$\frac{d\hat{u}_k}{dt} + \sum_{p+q=k} \hat{u}_p iq\hat{u}_q = \frac{-1}{Re}k^2\hat{u}_k + F_k, \text{ where } k \in \{-N, ..., N\} \tag{5.6}$$

In this point, let us make two important comments:

1. As $u(x,t) \in \mathbb{R}$ and $\hat{u}_k(t) \in \mathbb{C}$, then:

   $$\overline{\hat{u}_k(t)} = \hat{u}_{-k}(t) \ \ \forall k \in \mathbb{C}$$

   which represents an important property in terms of efficiency, given that we will only need to compute half of the Fourier coefficients, $\hat{u}_k$, and the other half will be immediately obtained by calculating their complex conjugate. Additionally, this property also results into an easier and faster way of obtaining the physical velocity:

   $$u(x,t) \simeq \sum_{k=-N}^{k=N} \hat{u}_k(t)e^{ikx} = \hat{u}_0(t) + \sum_{k=1}^{k=N} 2\,\mathrm{Re}\left(\hat{u}_k(t)e^{ikx}\right) \tag{5.7}$$

2. All along the previous chapters, we have imposed all boundary conditions in a direct and physically meaningful manner. However, as we are now going to run practically all the calculations in the Fourier space, this aspect may become rather fuzzy. More concretely, boundary conditions will be imposed by means of the forcing term affecting directly the Fourier coefficients and, implicitly, all

physical quantities. In particular, we are going to impose the following "boundary conditions":

$$F_k = \begin{cases} \sum_{p+q=k} \hat{u}_p iq\hat{u}_q + \frac{1}{Re}k^2\hat{u}_k & \text{if } k=1 \\ 0 & \text{otherwise} \end{cases} \tag{5.8}$$

Note that this definition implies that:

$$\frac{d\hat{u}_1}{dt} = 0 \implies \hat{u}_1 \equiv \text{ct.}$$

Going back to Equation 5.6 and integrating it over a certain time-step Δt gives:

$$\int_{\Delta t} \frac{d\hat{u}_k}{dt} dt = \int_{\Delta t} \left(-\sum_{p+q=k} \hat{u}_p iq\hat{u}_q - \frac{1}{Re}k^2\hat{u}_k + F_k \right) dt$$

and, considering a fully explicit scheme together with Equation 5.8, we can reach the following expression:

$$\hat{u}_k^{n+1} = \begin{cases} \hat{u}_k & \text{if } k=1 \\ \hat{u}_k\left(1 - \frac{\Delta t}{Re}k^2\right) - \Delta t C\hat{o}nv_k & \text{otherwise} \end{cases} \tag{5.9}$$

where the convective term satisfies the following:

$$C\hat{o}nv_k = \sum_{p+q=k} \hat{u}_p iq\hat{u}_q$$

Finally, let us extend all the preceding results to the general Burgers equation. In that case, Equation 5.4 becomes:

$$\mathbf{v}(x,t) \simeq \sum_{\|\mathbf{k}\| \leq N} \hat{\mathbf{u}}_{\mathbf{k}}(t) e^{i\mathbf{k}\cdot\mathbf{r}} \tag{5.10}$$

Equation 5.6:

$$\frac{d\hat{\mathbf{u}}_{\mathbf{k}}}{dt} + \mathbf{C\hat{o}nv}_{\mathbf{k}} = -\frac{\|\mathbf{k}\|^2}{Re}\hat{\mathbf{u}}_{\mathbf{k}} + \mathbf{F}_{\mathbf{k}}, \text{ where } \|\mathbf{k}\| \leq N \tag{5.11}$$

with:

$$\mathbf{C\hat{o}nv}_{\mathbf{k}} = \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} \left(\hat{\mathbf{u}}_{\mathbf{p}} \cdot i\mathbf{q}\right)\hat{\mathbf{u}}_{\mathbf{q}}$$

and Equation 5.9 results into:

$$\hat{\mathbf{u}}_{\mathbf{k}}^{n+1} = \begin{cases} \hat{\mathbf{u}}_{\mathbf{k}} & \text{if } \|\mathbf{k}\| \in [0.5, 1.5) \\ \hat{\mathbf{u}}_{\mathbf{k}}\left(1 - \frac{\Delta t}{Re}\|\mathbf{k}\|^2\right) - \Delta t\mathbf{C\hat{o}nv}_{\mathbf{k}} & \text{otherwise} \end{cases} \tag{5.12}$$

Spectral resolution

The recursive application of Equation 5.12 represents a fully explicit algorithm that, given Re and $\hat{\mathbf{u}}_{\mathbf{k}}(t=0)$, would allow us to obtain the distribution of Fourier coefficients and physical velocities of the system at the stationary state. As all the calculations take place in the Fourier space, this methods are known as spectral.

In spite of its simplicity, this resolution strategy is not perfect, given that the computation of the convective term may result very heavy depending on the dimension of the case being studied. It is enough to have a look at its expression to realize it:

$$\hat{\mathbf{Conv}}_{\mathbf{k}} = \begin{cases} \sum_{p+q=k} \hat{u}_p i q \hat{u}_q & 1D \\ \sum_{p_1+q_1=k_1} \sum_{p_2+q_2=k_2} \left(\hat{\mathbf{u}}_{\mathbf{p}} \cdot i\mathbf{q}\right) \hat{\mathbf{u}}_{\mathbf{q}} & 2D \\ \sum_{p_1+q_1=k_1} \sum_{p_2+q_2=k_2} \sum_{p_3+q_3=k_3} \left(\hat{\mathbf{u}}_{\mathbf{p}} \cdot i\mathbf{q}\right) \hat{\mathbf{u}}_{\mathbf{q}} & 3D \end{cases} \tag{5.13}$$

Consequently, the application of spectral methods would be really poor in terms of computational cost if the dimension is higher than one, given that the summation involved in the convective term becomes bigger and bigger.

Pseudo-spectral resolution

As has just been seen, the application of spectral methods may lead to really high computational cost, specially if high accuracy is sought. In order to overcome this situation, there has also been implemented a slight modification of the spectral methods known as pseudo-spectral.

The main idea of the pseudo-spectral method that we have utilized consists of calculating the convective term in the physical space, skipping the heavy summations of Equation 5.13. Therefore, it will be calculated as usual:

$$\mathbf{Conv} = (\mathbf{v} \cdot \nabla)\, \mathbf{v}$$

When it comes to the spatial partial derivatives involved in the convective term, they could be approximated in the physical space making use of the approximations explained in Part I. However, it is possible to obtain a higher accuracy taking advantage of Fourier coefficients and applying the following equalities:

$$\hat{\left(\frac{\partial \mathbf{v}}{\partial r_\mu}\right)}_{\mathbf{k}} = i k_\mu \hat{\mathbf{u}}_{\mathbf{k}}, \text{ where } \mu \in \{1,2,3\}$$

One final remark to be made is that, as explained in [8], in order to avoid aliasing errors arising from the non-linearity of the convective term, there has been applied the 3/2-rule, which consists of working with a 3/2 times finer mesh than desired. Additionally, there will be set to zero all the Fourier coefficients such that any of the components of its frequency, $\mathbf{k}$, lies within the interval $(N, 3N/2]$. This action will prevent any aliasing error coming from the products made in the physical space to calculate the convective term.

As a summary of this method, a diagram summarizing its steps per cycle may be found in Figure 5.1.

$$\begin{tikzcd}
\boxed{\textbf{Fourier space}} & \boxed{\textbf{Physical space}} \\
\hat{\mathbf{u}}_{\mathbf{k}} \arrow[r] & \mathbf{v} \\
\left(\frac{\widehat{\partial \mathbf{v}}}{\partial r_\mu}\right)_{\mathbf{k}} \arrow[r, "\text{de-alising}"] & \frac{\partial \mathbf{v}}{\partial r_\mu} \arrow[d] \\
\widehat{\mathbf{Conv}}_{\mathbf{k}} \arrow[u] & \mathbf{Conv} = (\mathbf{v} \cdot \nabla)\mathbf{v} \arrow[l]
\end{tikzcd}$$

Figure 5.1: Diagram showing the steps to be followed recursively in the resolution, using pseudo-spectral methods, of Burgers equation.

5.2 Results

In this section there will be shown some particular results arising from the implementation of the algorithm developed so far and applied to the one and two-dimensional cases. All results assume the following initial distribution of the Fourier coefficients:

$$\hat{\mathbf{u}}_{\mathbf{k}}(t=0) = \begin{cases} 0 & \text{if } k=0 \\ \\ {}^{1}/_{\|\mathbf{k}\|} & \text{otherwise} \end{cases} \tag{5.14}$$

We will now show the results obtained with two different type of implementations. First two subsections will deal with one and two-dimensional Burgers equation resolved using spectral methods, i.e., implementing Equation 5.12. On the other side, last subsection will present the results of the two-dimensional case resolved by means of pseudo-spectral methods, which represent an improvement in efficiency and will be properly detailed there.

1D Burgers Equation

First of all, let us show the periodic distribution of the velocity for different values of N. As is clearly seen in Figure 5.2, too low values of N leave to inaccurate approximations of the physical velocity, which is quite obvious bearing in mind the approximation made in Equation 5.4.

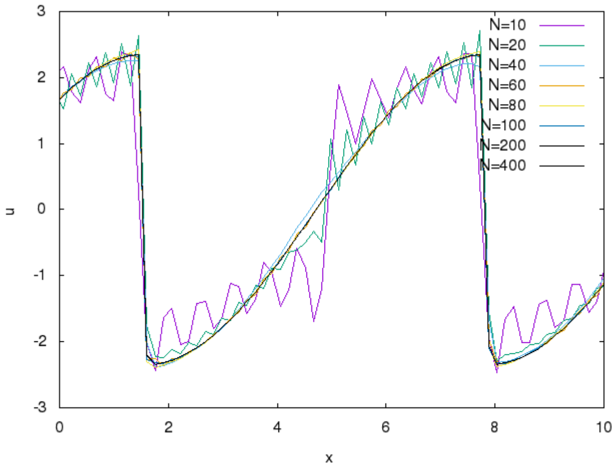


Figure 5.2: Distribution of the velocity energy in terms of the position for different values of N. Results obtained using $Re = 40$.

A really interesting phenomenon related to turbulence consists on the transport of energy from small frequencies to the high ones or, equivalently, from large scales to small scales, where energy is dissipated. This results into what is called energy cascade (see Figure 5.3) and somehow implies that, in order to reach reasonable results, N needs to be small enough so that the energy gets dissipated realistically.

With the aim of illustrating this behavior, kinetic energy ($E_k = \hat{u}_k \overline{\hat{u}_k}$) has been plotted in Figure 5.3, clearly showing the energy cascade and the importance of considering a proper value of N, even if this has a negative impact on the computational cost of the simulation.

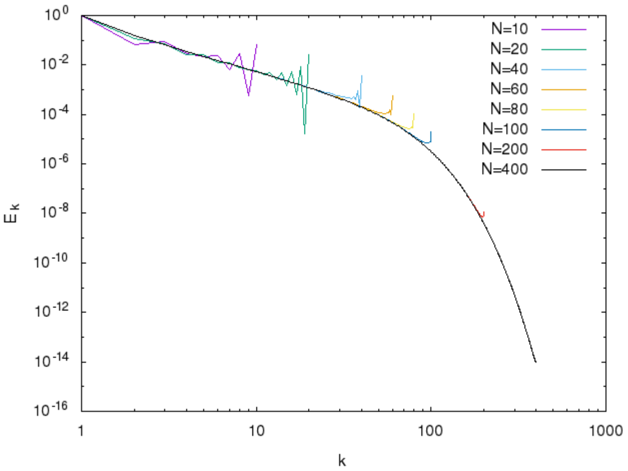
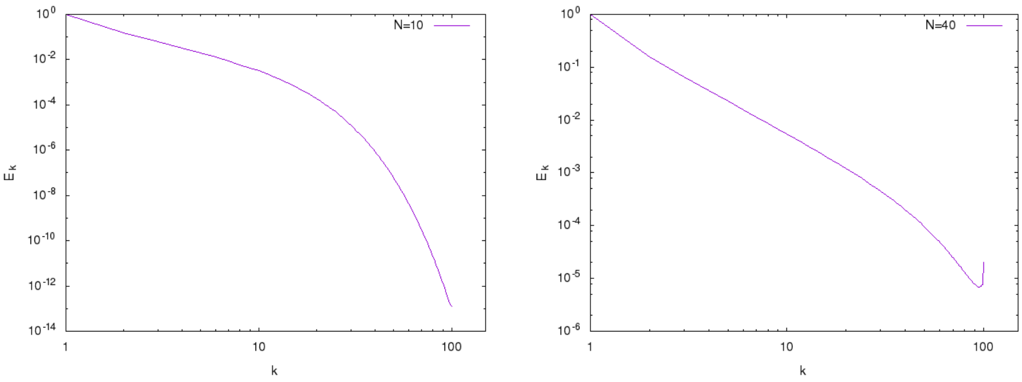


Figure 5.3: Distribution of kinetic energy in terms of the frequency for different values of N. Results obtained using $Re = 40$.

It is also interesting to see the impact of considering different Reynolds numbers on the results. One could think that it is enough to take $N = 100$, given that, as can be seen in Figures 5.2 and 5.3, the results are totally accurate regarding our purposes. However, a higher value of Re implies a need for considering smaller scales, i.e. bigger values of N, in order to reach the same accuracy. This fact is clearly shown in Figure 5.3, where $Re > 40$ and $N = 100$ leads to totally unrealistic results, whereas $Re < 40$ and $N = 100$ represents an unnecessarily high resolution.

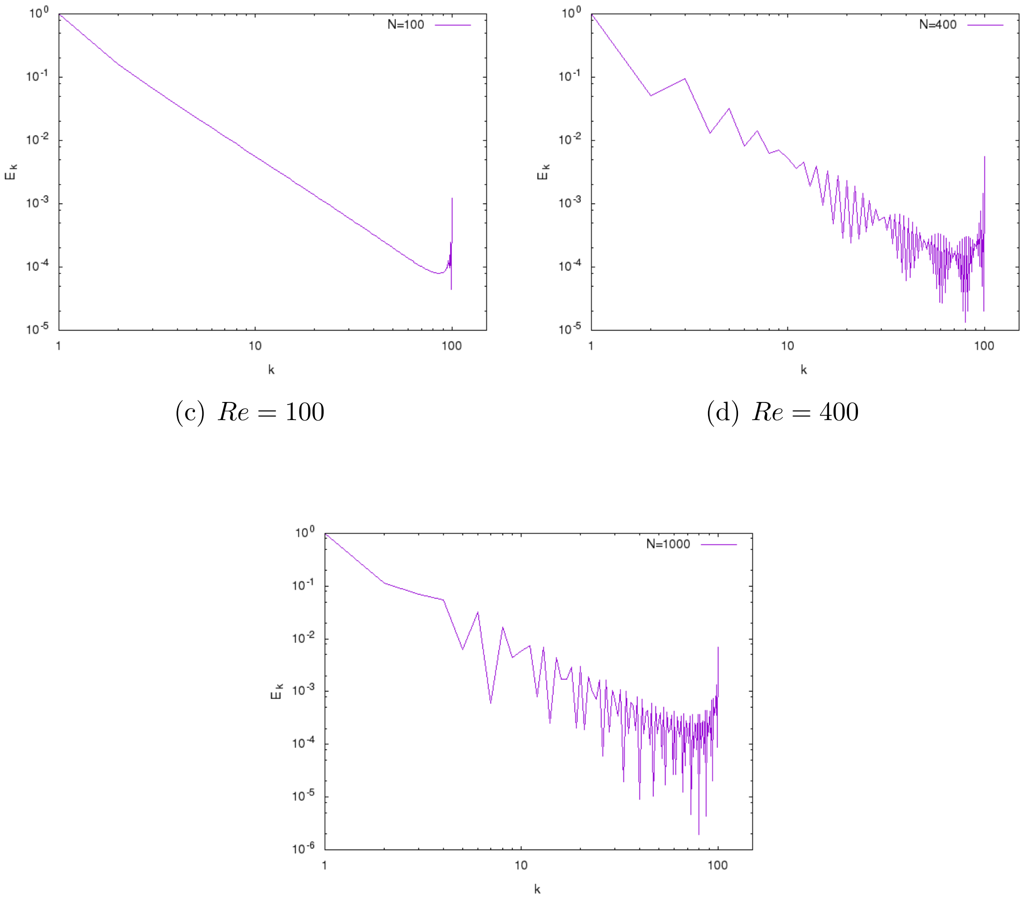
Precisely for this, the cost of running a DNS (direct numerical simulation) for high values of Re easily degenerates into an unaffordable computing cost.

As a final conclusion, it is interesting to note that from all these questions arises the need for modelling small scales (among other strategies) in order to reach reasonable results in reasonable times.



(a) $Re = 10$

(b) $Re = 40$



(c) $Re = 100$

(d) $Re = 400$

(e) $Re = 1000$

Figure 5.3: Energy cascades corresponding to different values of Re. Results obtained using $N = 100$.

2D Burgers Equation

In order to compare the huge differences in terms of efficiency between spectral and pseudo-spectral methods, we have run the same case with both implementations, reaching totally conclusive results. It is worth to emphasize that the simulation based on pseudo-spectral methods makes use of the *FFTW* library (see [9]), which is highly optimized and allows us to reach an even higher performance.

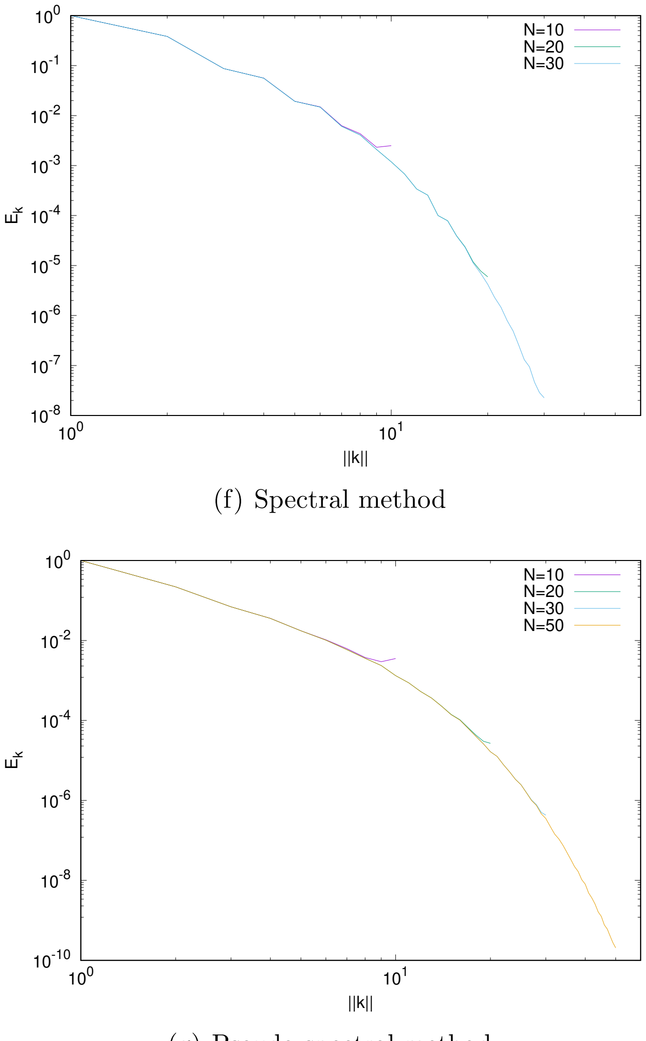
Hence, the results of comparing the execution time per iteration for different meshes has been summarized in Table 5.1. As expected, higher meshes (implying heavier computations) imply higher differences in performance.

Finally, energy cascades have been plotted for each implementation in Figure 5.4. The same results as for the one-dimensional case have been obtained. In this cases kinetic energy has been calculated as follows:

$$E_k = \sum_{||\mathbf{k}|| \in [k-0.5, k+0.5)} \hat{\mathbf{u}}_{\mathbf{k}} \overline{\hat{\mathbf{u}}_{\mathbf{k}}}, \text{ where } k \in \{0, \cdots, N\}$$

N	**Spectral method**	**Pseudo-spectral method**
10	9.27ms	0.151ms
20	140ms	0.894ms
30	688ms	0.981ms
50	5.17s	9.54ms
100	80.1s	22.2ms

Table 5.1: Comparison of the execution time per iteration depending on the method being used.Results obtained normalizing the times corresponding to 100 iterations.



(f) Spectral method

(g) Pseudo-spectral method

Figure 5.4: Distribution of kinetic energy in terms of the frequency for different values of N. Results obtained using $Re = 10$.

Chapter 6

Homogeneous Isotropic Turbulence

This chapter contains the last simulation that has been developed. More concretely, we are going to solve Navier-Stokes equation and study the phenomenon of homogeneous isotropic turbulence. In spite of the fact that the complexity of this case is considerably high, we will take advantage of the results obtained all along chapters 3, 4 and 5.

Our aim will consist of simulating periodic solutions of Navier-Stokes equation. These simulations will be implemented together with a turbulence model that will improve the solutions obtained with coarse meshes (compensating the lack of dissipation associated to smaller scales).

In particular, we will implement the Smagorinsky-Lilly model as presented in [11–13]. Additionally, we will adapt our code to study the decay of isotropic turbulence when no energy is being injected to the system and use the bench marks presented by A. A. Wray in [14] (case HOM02) to validate our program and adjust the constants of the turbulence model to that case. Finally, if the reader is not familiar with tensor operations, revising the summary presented in [10] may be very helpful.

6.1 Discretization

Incompressible Navier-Stokes equation in primitive variables reads:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{v} \tag{6.1}$$

Large-eddy simulations (from now on LES) arise from filtering Equation 6.1 using a spatial filter of a certain characteristic length δ (we will use $\delta = \sqrt[3]{V_{CV}}$). Thus, velocity is split into two components: filtered ($\bar{\mathbf{v}}$) and sub-filtered ($\mathbf{v}'$) velocities. Thus,

$$\mathbf{v} = \bar{\mathbf{v}} + \mathbf{v}' \tag{6.2}$$

Following the development of LES models made in [11], applying the filtered decomposition of Equation 6.2 results into the following modified version of the Navier-Stokes equation:

$$\frac{\partial \bar{\mathbf{v}}}{\partial t} + (\bar{\mathbf{v}} \cdot \nabla)\bar{\mathbf{v}} = -\frac{1}{\rho}\nabla \bar{p} + \nu \Delta \bar{\mathbf{v}} - \nabla \cdot \tau(\bar{\mathbf{v}}) \tag{6.3}$$

where $\tau(\bar{\mathbf{v}})$ is referred as subgrid-stress tensor and, under the eddy-viscosity assumption, satisfies:

$$\tau(\bar{\mathbf{v}}) \simeq -2\nu_e S(\bar{\mathbf{v}}), \text{ where } \begin{cases} \text{Eddy-viscosity:} & \nu_e \\ \text{Rate-of-strain tensor:} & S(\bar{\mathbf{v}}) = {}^1\!/\!{}_2\left(\nabla\bar{\mathbf{v}} + \nabla\bar{\mathbf{v}}^t\right)) \end{cases} \tag{6.4}$$

Additionally, if we consider the Smagorinsky model, then we can assert that:

$$\nu_e = (C_S \delta)^2 \left|S(\bar{\mathbf{v}})\right|, \text{ where } \left|S(\bar{\mathbf{v}})\right| = \sqrt{(2S : S)} \tag{6.5}$$

where the Smagorinsky constant, C_S, should be tuned adequately (bearing in mind that, theoretically, $C_S \simeq 0.17$).

Bearing in mind the development of the FSM done in chapters 3 and 4, one could easily realize that the predictor velocity that was being calculated there coincides with the velocity obtained resolving the Burgers equation as in Chapter 5 (for the periodic solution). Therefore, both methods could be combined to find the periodic solution of Equation 6.3.

It is worth to mention that, in the current case, the method detailed in Chapter 5 should be reformulated in order to include the modifications due to the application of the LES model. Additionally, in spite of the fact that throughout this chapter we will be dealing with the filtered velocity $\bar{\mathbf{v}}$, we will refer to it as $\mathbf{v}$ in order to simplify all notations. Thus, the recursive formula to obtain the predictor velocity (in the Fourier space) would be:

$$\hat{\mathbf{u}}^p_{\mathbf{k}} = \hat{\mathbf{u}}_{\mathbf{k}} \left(1 - \frac{\Delta t}{Re}\|\mathbf{k}\|^2\right) - \Delta t \hat{\mathbf{Conv}}_{\mathbf{k}} - \nabla \cdot \tau_{\mathbf{k}}(\mathbf{v}) \tag{6.6}$$

where $\hat{\mathbf{u}}^p_{\mathbf{k}}$ refers to the predictor velocity of the FSM and the divergence of the subgrid-stress tensor could be understood as playing the role of the forcing term.

In this point, predictor velocity should be projected to a divergence-free subspace. Similarly to what we had in Equation 3.5, in the Fourier space we would have:

$$\Delta \hat{\tilde{p}}_{\mathbf{k}} = \nabla \cdot \hat{\mathbf{u}}^p_{\mathbf{k}} \tag{6.7}$$

which could easily be resolved, leading to:

$$\hat{\hat{p}}_{\mathbf{k}} = -i\frac{\mathbf{k}\cdot\hat{\mathbf{u}}^p_{\mathbf{k}}}{\|\mathbf{k}\|^2}, \text{ for a given Fourier frequency } \mathbf{k} \tag{6.8}$$

and, consequently,

$$\nabla\hat{\hat{p}} \simeq \sum_{\|\mathbf{k}\|\leq N} i\mathbf{k}\hat{\hat{p}}_{\mathbf{k}}e^{i\mathbf{k}\cdot\mathbf{r}} = \sum_{\|\mathbf{k}\|\leq N} \frac{\mathbf{k}\cdot\hat{\mathbf{u}}^p_{\mathbf{k}}}{\|\mathbf{k}\|^2}\mathbf{k}e^{i\mathbf{k}\cdot\mathbf{r}} = \sum_{\|\mathbf{k}\|\leq N} \frac{\mathbf{k}^t\cdot\mathbf{k}}{\|\mathbf{k}\|^2}\hat{\mathbf{u}}^p_{\mathbf{k}}e^{i\mathbf{k}\cdot\mathbf{r}} \tag{6.9}$$

therefore, the analogue of Equation 3.9 would be:

$$\hat{\mathbf{u}}^{n+1}_{\mathbf{k}} = \hat{\mathbf{u}}^p_{\mathbf{k}} - \nabla\hat{\hat{p}}_{\mathbf{k}} \tag{6.10}$$

and, combined with Equation 6.9,

$$\hat{\mathbf{u}}^{n+1}_{\mathbf{k}} = \left(\mathbb{I} - \frac{\mathbf{k}^t\cdot\mathbf{k}}{\|\mathbf{k}\|^2}\right)\hat{\mathbf{u}}^p_{\mathbf{k}} \tag{6.11}$$

where $\left(\mathbb{I} - \frac{\mathbf{k}^t\cdot\mathbf{k}}{\|\mathbf{k}\|^2}\right)$ plays the role of a projector to a divergence-free subspace.

6.2 Results

The implementation of all the formalism introduced in the last subsection has been implemented in two and three dimensions, leading to totally coherent results. However, in order to validate definitively our results, it may be interesting to contrast with a dataset obtained from literature.

With this aim, we have used the results provided in [14]. More concretely, we have loaded an initial set of velocities which corresponded to a case of isotropic turbulence. Then, we have let it evolve in absence of any injection of energy and, finally, we have stored its energy distribution at time $t \simeq 0.494s$. Finally, its correspondence with the given dataset has been checked using $N = 30$ for different values of C_S, which, as we said, should take a theoretical value equal to 0.17, even though it normally goes down to values around 0.11.

Effectively, these cases has been plotted in Figure 6.1, leading to the conclusion that the best value for C_S is 0.11, given that it adds the perfect amount of extra viscosity, dissipating the exact amount of kinetic energy.

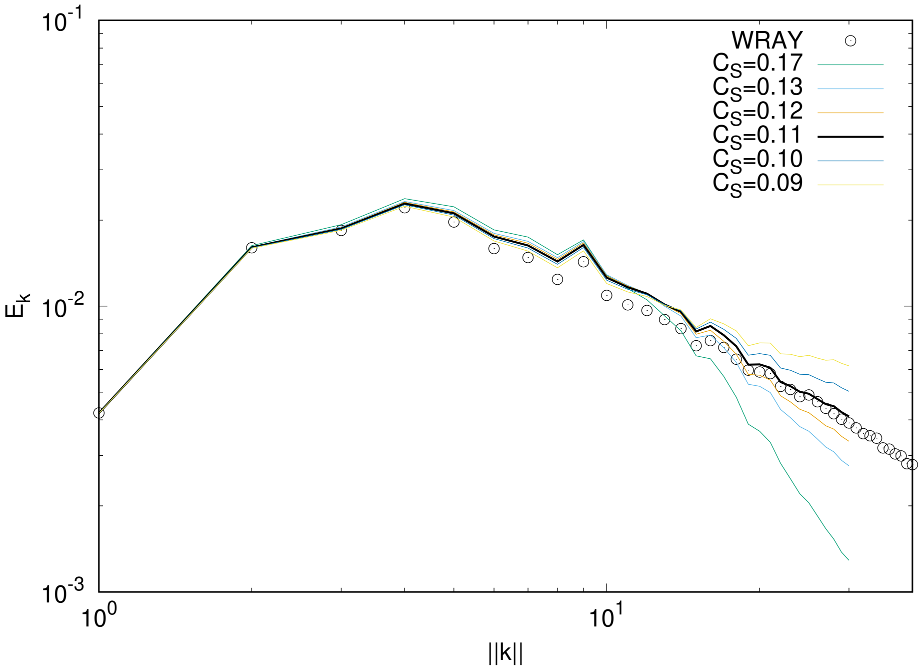


Figure 6.1: Distribution of kinetic energy in terms of the frequency for different values of C_S and compared to the benchmark (WRAY). Results obtained using $N = 30$ and $\nu = 3.283e - 4$.

Let us also note that, as three-dimensional isotropic turbulence is already a rather heavy case, we have implemented it in two ways, sequentially and in parallel. Its parallel version makes use of the library *Message passing interface* (MPI), which allows us to run the code by several CPUs concurrently, reducing significantly the total execution time. However, it has to be parallelized wisely, given that MPI is based on communications between processors. Therefore, too many communications could have a worse effect than running the program sequentially. In Table 6.1, there can be found a comparison of execution times per iteration for different implementations run with the same computer.

N	**Sequential code**	**2 CPUs**	**3 CPUs**	**4 CPUs**
10	17.2ms	12.6ms	11.6.ms	9.16.ms
20	227ms	163ms	121ms	109ms
30	313ms	333ms	340ms	341ms
50	4.67s	3.29s	4.28s	2.20s

Table 6.1: Comparison of the execution time per iteration depending on the method being used. Results obtained normalizing the times corresponding to 30 iterations.

Part III

Concluding remarks

Conclusions

After the study of these six cases of fluid dynamics and heat transfer, the potential utility of CFD to a wide range of sectors will, for sure, have become completely obvious to any reader. For example, it would make possible to avoid expensive and inaccurate experimental tests in the industrial sector, or it would enrich the knowledge that one may have about any sort of device where heat transfer may play a role.

However, considering our results, we can conclude that not only accuracy is decisive for a successful application of CFD, but also efficiency and performance (one just needs to remember the execution times of spectral versus pseudo-spectral methods).

On the one side, lower execution times would increase the range of applications of the simulations to fields where almost real-time results are needed. On the other side, this would also suppose lower power consumptions.

Precisely for the huge range of applications of CFD, saving as much computation power as possible has become a necessity in order to lower its indirect but more than considerable environmental impact.

Therefore, under these assumptions it makes a lot of sense to focus future research on the optimization of iterative codes such as the ones involved in CFD. More concretely, it would be really interesting to work in two complementary directions: firstly, looking for better discretizations and resolution methods and, secondly, adapting the simulations to parallel computing and optimizing them to the last and most efficient architectures.

Bibliography

[1] Patankar, S. V. (1980). Numerical Heat Transfer and Fluid Flow, *Hemisphere Publishing Corporation.*

[2] Pérez-Segarra, C.D., Farre, C., Cadafalch, J. and Oliva, A. (2006). Analysis of Different Numerical Schemes for the Resolution of Convection-Diffusion Equations using Finite-Volume Methods on Three-Dimensional Unstructured Grids. Part I: Discretization Schemes. *Numerical Heat Transfer*, Part B, 49(4), 333-350.

[3] Unstructured finite-volume discretization of the convection- diffusion equation (2013) [unpublished], Centre tecnològic de transferència de calor (CTTC), Terrassa.

[4] Introduction to the Fractional Step Method (2013) [unpublished], Centre tecnològic de transferència de calor (CTTC), Terrassa.

[5] de Vahl Davis, G. and Jones, I. P. (1983). Natural convection in a square cavity: a comparison exercise, *International Journal for numerical methods in fluids*, 3, 227-248.

[6] de Vahl Davis, G. (1983). Natural convection of air in a square cavity bench mark numerical solution, *International Journal for numerical methods in fluids*, 3, 249-264.

[7] Burgers equation in Fourier space (2014) [unpublished], Centre tecnològic de transferència de calor (CTTC), Terrassa.

[8] Lu, F. (2017). De-aliasing in Fast Fourier Transform. Retrieved from `http://www.math.jhu.edu/~feilu/notes/DealiasingFFT.pdf`.

[9] Documentation of FFTW 3.3.8. Retrieved from `http://www.fftw.org/fftw3_doc/`.

[10] Kolecki, J. C. (2002). An Introduction to Tensors for Students of Physics and Engineering. Retrieved from `https://www.grc.nasa.gov/www/k-12/Numbers/Math/documents/Tensors_TM2002211716.pdf`.

[11] Large-Eddy Simulation of turbulent incompressible flows (2015) [unpublished], Centre tecnològic de transferència de calor (CTTC), Terrassa.

[12] Trias, F.X., Folch, D., Gorobets, A. and Oliva, A. (2015). Building proper invariants for eddy-viscosity subgrid-scale models, *Physics of fluids*, Physics of Fluids, 27(6):065103.

[13] Trias, F.X., Gorobets, A., Oliva, A. and Verstappen, R.W.C.P. (2015). New Differential Operators for Large-Eddy Simulation and Regularization Modeling. In *Direct and Large-Eddy Simulation IX*, 29-35.

[14] Wray, A. A. (1998). A Selection of Test Cases for the Validation of Large-Eddy Simulations of Turbulent Flows, *AGARD advisory report*, 345.