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

Symmetry-preserving discretizations applied to Large-Eddy Simulation techniques in Navier-Stokes equations

Bachelor's Thesis - Report

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Escola Superior d'Enginyeria Industrial, Aeroespacial i Audiovisual de Terrassa

Author:
Plana Riu, Josep

Director:
Pérez Segarra, Carlos-David
Co-director:
Oliva Llena, Asensio

DELIVERY DATE: September 28th 2020

Treball entregat per a l'obtenció de: *Graduat en Enginyeria en tecnologies aeroespacials per la Universitat Politècnica de Catalunya*

Acknowledgements

Primer de tot, m'agradaria donar les gràcies al meu director del TFG, el Carles-David, així com també al codirector, l'Assensi, per posar a la meva disposició el que requerís en tot moment, així com també per haver-me obert les portes del CTTC, juntament amb el Quim, el desembre del 2018 i així poder endinsar-me en aquest món tan apassionant.

També m'agradaria agrair especialment el suport en tot moment durant l'elaboració d'aquest treball per part del Xavi: ja sigui des del punt de vista dels solvers lineals, fins a la teoria de la turbulència com en l'aplicació dels models LES als casos pràctics, ha estat una part fonamental en l'elaboració d'aquest treball.

En tercer lugar, me gustaría dedicar unas palabras para Pablo, quien me ayudó muchísimo a dar mis primeros pasos en esto de los métodos numéricos, desde el simple caso de transferencia de calor por conducción hasta los casos laminares de Navier-Stokes. Muchísimas gracias por las charlas y los consejos en el despacho del laboratorio.

També voldria agrair els bons moment que he passat durant aquests quatre anys a tots els companys, però especialment al Lau, al Marc, al Pol, al Xordi i al Vlad, per sempre ser allà quan feia falta, en els bons i en els mals moments, però sobretot per aguantar el meu humor absurd durant tant de temps.

Finalment, i no per això menys important, voldria agrair de tot cor el suport de la meva família i amics, que moltes vegades es pensen que estic mig tarat per dedicar-me a aquestes coses, i ja ni intenten saber de què va el tema, però que malgrat això no han deixat en cap moment d'ajudar en tot el que fes falta.

Abstract

Direct Numerical Simulations of the incompressible Navier-Stokes equations for relatively high Reynolds numbers, as required for airfoils, are extremely expensive in terms of number of CPUs as well as processing time. Thus, small-scale modelling is a clever way to reduce this cost by introducing an extra dissipation in the form of a turbulent viscosity.

In this thesis, the turbulence phenomenon is reviewed, from both theoretical and technical points of view, and applied to a turbulent Lid-Driven Cavity. In order to do so, different eddy-viscosity models are applied in a Large Eddy Simulation formulation melded into a symmetry-preserving discretization that presents the optimal conditions for turbulence simulation.

In fact, S3PQ model developed by CTTC is used and tested in a Lid-Driven Cavity at $\mathrm{Re} = 10000$, provided its remarkable turbulent properties, in which the properties of a turbulent boundary layer are studied and compared to the theoretical approach, previously developed.

List of Symbols

Abbreviations

CDS	Central Difference Scheme
CFD	Computational Fluid Dynamics
CG	Conjugate Gradient Method
DHC	Differentially Heated Cavity
DNS	Direct Numerical Simulation
EoM	Equations of motion
FSM	Fractional Step Method
LDC	Lid Driven Cavity
LES	Large-Eddy Simulation
NS	Navier-Stokes equations
PDF	Probability density function
SIMPLER	Semi-Implicit Method for Pressure-Linked Equations Revised

Operators

$(\cdot) \star (\cdot)$	Convolution product
$\hat{(\cdot)}$	Fourier space transformation
$\langle\cdot\rangle$	Time-averaging
$\mathrm{G}(\cdot)$	Convolution kernel
$\tilde{(\cdot)}$	Filtered function

Tensorial symbols

δ_{ij}	Kronecker's symbol
e_i	i-th component of the reference frame

Non-dimensional numbers

Pe	Péclet number
Pr	Prandtl number

Ra	Rayleigh number
Re	Reynolds number

Non-dimensional magnitudes

$\hat{\delta}$	Dimensionless boundary layer height
$\hat{p}$	Non-dimensional pressure field
$\hat{t}$	Non-dimensional time
$\hat{u}_i$	i-th component of the non-dimensional velocity field
$\hat{x}_i$	i-th component of the non-dimensional position

Physical properties

α	Thermal diffusivity
β	Thermal expansion coefficient
η	Kolmogorov's length scale
Γ	Diffusivity associated to an arbitrary scalar
κ	Thermal conductivity
λ	Thermal conductivity
μ	Viscosity
ν	Kinematic viscosity (μ/ρ)
ρ	Density
C_p	Specific heat at constant pressure

Subindices

η	Scale associated to Kolmogorov's hypotheses
∞	Reference value
w	Magnitude calculated at the wall

Physical variables

δ	Boundary layer height
$\mathcal{L}$	Length scale for non-dimensioning
$\mathcal{U}$	Velocity scale for non-dimensioning
ϕ	Arbitrary scalar
Φ^e	Dissipated energy by Joule effect
τ	Eddy-timescale
τ	Friction
ε	Rate of dissipation

c_f	Skin-friction coefficient
C_{ij}	Cross-stress tensor
e	Internal energy
f_i	i-th component of the body forces
g_i	i-th component of the gravitational acceleration
L_{ij}	Leonard tensor
p	Pressure field
R_{ij}	Reynolds subgrid tensor
T	Temperature field
T_{ij}	Subgrid stress tensor
u'_i	i-th component of the perturbations of the velocity field
u_τ	Friction velocity
u_i	i-th component of the velocity field
x_i	i-th component of the cartesian reference frame
y^+	Normal distance to the wall (wall units)
k	Turbulent kinetic energy

List of Tables

List of Figures

List of Algorithms

Contents

Honor declaration

I declare that,

the work in this Degree thesis is completely my own work,

no part of this Degree thesis is taken from other people's work without giving them credit,

all references have been clearly cited,

I am authorised to make use of CTTC related informaton I am providing in this document,

I understand that an infringement of this declaration leaves me subject to the foreseen disciplinary actions by the Universitat Politècnica de Catalunya - Barcelona Tech

Josep Plana Riu, September 28th 2020

Title of the thesis: Symmetry-preserving discretizations applied to Large-Eddy Simulation techniques in Navier-Stokes equations

Chapter 1

Introduction

1.1 Aim of the thesis

The aim of the thesis is, ultimately, to properly simulate the phenomenon of turbulence in fluids. Nonetheless, in order to be able to provide the expected results, which have to be physically coherent, a solid background on turbulence has to be built, from both a theoretical point of view and a computational point of view.

The objectives of this thesis are, thus, the following:

1. Understand how turbulence works and how it is modelled.
2. Implement a C code applying the Finite Volume Method to solve reference cases using LES models.

1.2 Background and justification

Turbulence is not a well known phenomena. As 1965 Nobel Prize in Physics winner Richard Feynman stated in his Lectures on Physics [1]: *there is a physical problem that is common to many fields, that is very old, and that has not been solved. It is not the problem of finding new fundamental particles, but something left over from a long time ago—over a hundred years. Nobody in physics has really been able to analyze it mathematically satisfactorily in spite of its importance to the sister sciences. It is the analysis of circulating or turbulent fluids.*

In fact, from a theoretical point of view, the analysis of turbulent flows has not progressed much since Feynman stated that in 1964: Kolmogorov had previously published its statistical and energetical analysis of the turbulence, and Reynolds and Prandtl had developed their own theories around Navier-Stokes equations.

This is one of the reasons why the proof of existence and smoothness of a solution to the incompressible Navier-Stokes equations in three dimensions, which detail the fluid motion, is one of the Millenium Prize problems.

Nevertheless, with the time passing by, the computational power of CPUs has increased quite a lot. Moreover, discretizing the Navier-Stokes equations both in space and time had been proven possible, which made them suitable to be solved in a computer. Thus, different methodologies have been developed and improved each other in order to obtain more accurate simulations as well as more complex.

This is the world of CFD, based on discretizing a domain in time and space so as to solving in each of the control volumes obtained an equivalent integral, in case of a Finite Control Volume approach, version of the Navier-Stokes equations, which, all summed up, will contribute to the solution of the full domain.

However, Kolmogorov's studies shown an increasingly dependance of the relevant scales of the fluid with how turbulent the flow was. Thus, for strongly turbulent flows, discretization scales for a DNS required to be too small in some problems, requiring an enormous computational power which was either unavailable or unfeasible. Thus, turbulence models were developed so that the scales could be slightly bigger and, therefore, the simulations were cheaper in terms of computational power.

This is where LES appears, being one of the most commonly used methods of turbulence modelling, which, in fact, might require some development in the near future to make it even cheaper to simulate the Navier-Stokes equations in detail.

1.3 Scope of the thesis

- Understanding the fundamentals of the turbulence theory from Reynolds, Prandtl and Kolmogorov, among others.
- Implementing the theory as well as simulation techniques such as the Large-Eddy Simulation in order to solve turbulent cases using the Finite Volume Method.
- Solving the Burgers' equation so as to get deeper on the energy cascade concept presented by Kolmogorov.
- Implement a staggered Navier-Stokes solver using the Fractional Step Method to solve some reference cases in both 2D and 3D.
- Testing different LES models in a reference case to determine which has the best properties and interacts less with the fluid scales that do not need to be modified.
- Performing a turbulent simulation using a LES model in my own laptop.

Chapter 2

General features and state-of-the-art

2.1 General features of Fluid Dynamics

In order to fully understand the phenomenon of turbulence, Fluid Mechanics is the most important field in physics to consider. Thus, a brief introduction to this field is intended in this brief section.

As any physical behaviour, fluids follow the conservation of different properties: in the general case; mass, momentum and energy.

2.1.1 Mass conservation

The conservation of mass is denoted by the continuity equation (2.1).

$$\frac{\partial \rho}{\partial t} + \rho \frac{\partial u_i}{\partial x_i} = 0 \tag{2.1}$$

In the case of an incompressible flow, ρ does not depend on time. Thus, (2.1) is modified to (2.2), which corresponds to a divergence-free velocity field.

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{2.2}$$

2.1.2 Momentum conservation

The conservation of momentum corresponds to (2.3).

$$\rho \left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \right) = -\frac{\partial p}{\partial x_i} + \mu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i \tag{2.3}$$

Generally the body forces f_i correspond to the weight of the fluid element (ρg_i), even though it can include all sorts of electromagnetic forces (e.g. Lorentz force). Now, it is here where some approximations can be performed at this term. It is typical in the case of buoyancy the use of Boussinesq approximation, which replaces the density at the gravitational term by (2.4), yielding (2.5).

$$\rho(T) = \rho_\infty - \rho_\infty \beta (T - T_\infty) \tag{2.4}$$

$$\rho \left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \right) = -\frac{\partial p}{\partial x_i} + \mu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + (1 - \beta(T - T_\infty))\rho_\infty g_i \tag{2.5}$$

2.1.3 Energy conservation

Following a convection-diffusion equation for energy e and considering that $e = C_p T$, the conservation of energy yields (2.6).

$$\rho C_p \left(\frac{\partial T}{\partial t} + u_i \frac{\partial T}{\partial x_i} \right) = \lambda \frac{\partial^2 T}{\partial x_j \partial x_j} - p \frac{\partial u_i}{\partial x_i} + \Phi^e + \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \frac{\partial u_i}{\partial x_j} \tag{2.6}$$

Nevertheless, in most cases the pressure, Joule effect and viscosity dissipation can be neglected, just yielding (2.7)

$$\rho C_p \left(\frac{\partial T}{\partial t} + u_i \frac{\partial T}{\partial x_i} \right) = \lambda \frac{\partial^2 T}{\partial x_j \partial x_j} \tag{2.7}$$

2.1.4 Similarities and invariances of Navier-Stokes equations

As stated in [2], in order to develop theoretical and computational models in turbulence which follow a correct qualitative approach these invariances and similarities have to be considered.

In order to do so, a non-dimensional form for (2.3) has to be provided. Nonetheless, the body forces will be neglected for the sake of simplicity. Thus, considering the independent variables to be

$$\hat{x}_i = x_i / \mathcal{L}$$
$$\hat{t} = t\mathcal{U} / \mathcal{L}$$

and the dependent variables

$$\hat{u}_i = u_i / \mathcal{U}$$
$$\hat{p} = p / (\rho \mathcal{U}^2)$$

Thus, by replacing all the non-dimensional values onto (2.3), (2.8) will be derived.

$$\frac{\partial \hat{u}_i}{\partial t} + \hat{u}_j \frac{\partial \hat{u}_i}{\partial \hat{x}_j} = -\frac{\partial \hat{p}}{\partial \hat{x}_i} + \frac{\mu}{\rho \mathcal{U} \mathcal{L}} \frac{\partial^2 \hat{u}_i}{\partial \hat{x}_j \partial \hat{x}_j} \tag{2.8}$$

where the Reynolds number (2.9) can be defined, which indicates the ratio of inertial forces and viscous forces. Thus, the bigger the Reynolds number is, the most relevant will be the inertial forces in the behaviour of the fluid, and vice versa.

$$\mathrm{Re} = \frac{\rho \mathcal{U} \mathcal{L}}{\mu} \tag{2.9}$$

Reynolds number similarity

Two different cases (scalewise, materialwise, timewise, etc.) but with an exact Reynolds number will behave the same exact way in the non-dimensional domain, being this the reason why it can be said that the outcome of the case will be similar in behaviour, yet different on magnitude, given that (2.8) just depends on Re.

Time and space invariance

Moving the experiment in time and space will just imply a rescale of independent variables that will not affect in any way (2.8). Thus, time and space shifting will not imply any change in the flow's behaviour.

Rotational and reflectional invariance

A fixed rotation or a reflection (double π-rotation) of the reference frame (characterized by rotation tensor a_{ji}) will not imply any change in (2.3). Thus, as detailed in [2], this invariance will have a deeper meaning, given that the implication related to this invariance implies no bias in left-handed or right-handed rotation of the flow from the EoM point of view.

Time reversal

A time reversal can be characterized by changing the parity of time. Thus, $\hat{u}_i$ will also be affected, which implies a change of the sign of the viscous term, being this the reason why NS are no time invariant.

Galilean invariance

As all classical physics phenomena, NS are Galilean invariant, which implies that the behaviour will be similar for all different inertial frames of reference. Furthermore, NS in particular are also invariant in rectilinear acceleration non inertial reference frames, as derived in [2], based on changing the non-dimensioning process of pressure:

$$\hat{p} = \frac{p + \rho \hat{x}_i A_i}{\rho \mathcal{U}}$$

Frame rotation

Given a rotative reference frame, by introducing the derivative in a rotative frame of reference, the fact that the fictitious accelerations (centrifugal, Coriolis and angular acceleration) appear implies that the same procedure for the Galilean invariance cannot be applied, given that Coriolis acceleration is velocity-dependant and angular acceleration is time-dependant. Thus, the NS will not be material-frame indifferent.

Nonetheless, for a 2D case with a steady rotation, both Coriolis acceleration and angular acceleration vanish, allowing to absorb the centrifugal acceleration onto the pressure term and, then, in this exact case NS are material-frame indifferent.

Chapter 3

Previous work

In order to get used to programming codes in which the Finite Volume Method was used, being this the method used in all cases simulated for the thesis, some previous work had been done, which has gone from simple cases from heat conduction in solids to the solution of the Navier-Stokes equations in both hydrodynamics and with heat transfer.

Even though a brief summary of the work done will be developed in the next sections, the full reports are available in the Appendices (see Appendices A, B, C, D).

3.1 Heat Conduction equation

This case based on a 2D rod made of four different materials (Table A.1) was done so as to solve the heat diffusion equation (3.1).

$$\frac{\partial}{\partial x_i}\left(\lambda \frac{\partial T}{\partial x_i}\right) + \dot{q}_v = \rho c_p \frac{\partial T}{\partial t} \tag{3.1}$$

The domain to solve and the different materials can be seen in Figure A.1, while the boundary conditions applied are Table A.2.

Thus, after performing a centered nodes FVM discretization a diagonal-dominant pentadiagonal system of equations was obtained, which was solved using the Gauss-Seidel line-by-line algorithm (algorithm 1), which corresponds to a mix of the TDMA (see [3]) and the Gauss-Seidel algorithm , in which a TDMA is performed for each line considering the north and south values as known by using its last calculated value, and repeated until convergence.

Algorithm 1: Gauss-Seidel line-by-line.

```
while $\varepsilon > \delta$ do
    $j := 0$;
    while $j < N_y$ do
        $b^* := b + AN(:,j)x(:,j+1) + AS(:,j)x(:,j-1)$;
        TDMA$(A(:,j), x, b^*)$;
        $j := j + 1$;
    end
    $r := b - Ax$;
    $\varepsilon = \sqrt{r^T r}$
end
```

After validating the code, different results concerning the processing time, the time step or the grid size were obtained in order to understand how, for example, changing the tolerance of the solver or changing the number of control volumes would affect in obtaining the desired results or how long would take to end the simulation.

In terms of the temperature at the desired points, the results from Table A.5 were obtained, yielding Figure 3.1.

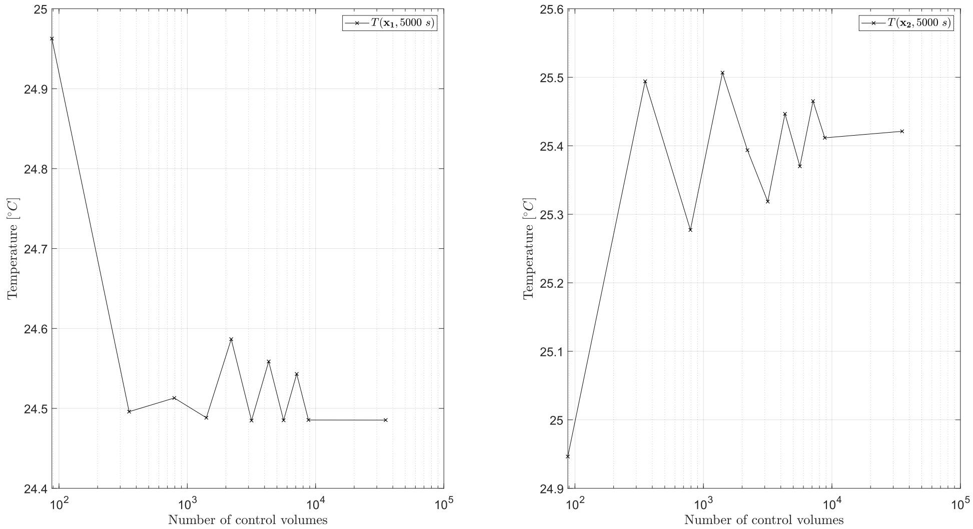


Figure 3.1: Mesh size convergence analysis for the case studied, with tolerance set at $\delta = 10^{-5}$ and a timestep of $\Delta_t = 1.0\ s$.

3.2 Convection-diffusion equation

After solving the 2D conduction case, the next step was getting introduced in the convection-diffusion equations (3.2), being both momentum and energy equations from Navier-Stokes examples of these. In these cases, the transport and diffusion of ϕ was studied knowing the flow field.

$$\frac{\partial(\rho\phi)}{\partial t} + \rho u_i \frac{\partial \phi}{\partial x_i} = \Gamma \frac{\partial^2 \phi}{\partial x_i \partial x_i} + S \tag{3.2}$$

Thus, different cases were conducted with analytical solution, with different Péclet numbers (3.3).

$$\text{Pe} = \frac{\rho u L}{\Gamma} \tag{3.3}$$

Nonetheless, the last case solved corresponded to the Smith-Hutton problem, defined by the boundary conditions from Table B.1 and the velocity field (3.4).

$$u(x,y) = 2y(1-x^2) \tag{3.4a}$$

$$v(x,y) = -2x(1-y^2) \tag{3.4b}$$

$$\tag{3.4c}$$

Thus, the equation was discretized using a FVM discretization. Nevertheless, in this case a simple discretization as presented in the conduction equation would not provide a stable algorithm. Thus, a general discretization can be obtained as (3.5), where F stands for any neighbour node, while f stands for the boundary of the node between F and P.

$$a_F = D_f A(|\mathrm{Pe}_f|) + \max\left(-n_f F_f, 0\right) \tag{3.5}$$

Moreover, here $D_f = \frac{\Gamma_f}{(\delta x_f |n_f|)}$ and $F_f = \rho u_f |n_f|$, whereas $A(|\mathrm{Pe}|)$ stands for the discretization scheme used to calculate the values at the boundaries. Even though multiple schemes can be used, in this case the PowerLaw scheme was used (3.6) [3].

$$A(|\mathrm{Pe}|) = \max\left(0, (1 - 0.1|\mathrm{Pe}|)^5\right) \tag{3.6}$$

After solving the system of equations using algorithm 1, the results for each ρ/Γ were obtained (see Table B.2-B.5), while the convergence analysis yields Figure 3.2.

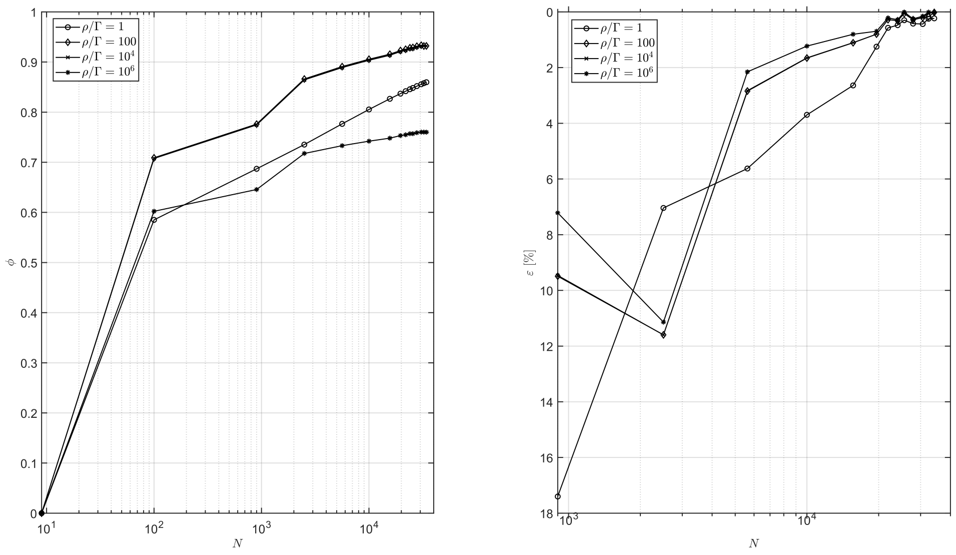


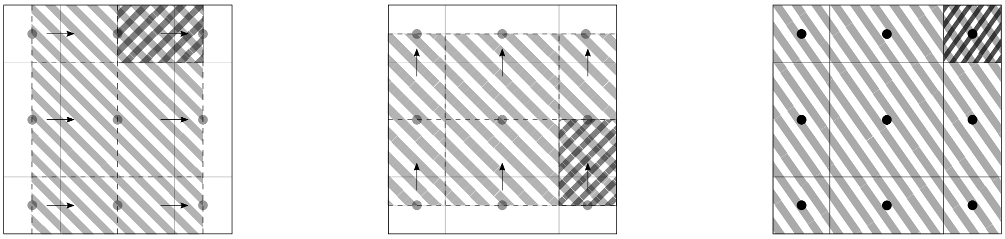
Figure 3.2: Convergence analysis for the Smith-Hutton Problem at different ρ/Γ numbers

3.3 Navier-Stokes equations

In order to numerically solve the Navier-Stokes equations, two different methods were used. First of all, following Patankar [3] methodology, as it seemed more natural given the previous cases, the SIMPLER (Semi-Implicit Method for Pressure-Linked Equations Revised) was used to solve one of the reference cases in CFD, the Lid Driven Cavity (LDC). Eventually, nonetheless, the Fractional Step Method [4] was used to solve both Lid Driven Cavity as well as the Differentially Heated Cavity (DHC) to conduct a case considering the energy equation.

3.3.1 SIMPLER Algorithm

First introduced by Patankar in 1979, the SIMPLER (algorithm 2) relies on the use of a staggered grid (Figure 3.3) to avoid the checkerboard problem, provided that the use of a non-staggered grid may lead to a case in which the pressure field has no physical sense.



(a) $x-$staggered grid. (b) $y-$staggered grid. (c) Main grid.

Figure 3.3: Representation of the staggered grid. In all representations, the dark cell corresponds to just a control volume, whereas the light cells correspond to all the domain. Extracted from [5].

Thus, the algorithm, which is a modified version of the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations), introduced by Patankar and Spalding back in 1972, is based on decoupling velocity and pressure by splitting the latter in two different fields p^* and p'.

In this case, the numerical scheme used to compute the velocity at the boundaries of the staggered grid was the Central Difference Scheme (CDS), even though it is slightly unstable, because its use is the most extended in the solution of turbulent flows.

The case solved was the LDC, which is described in Appendix C, for different Reynolds, being in this case Re = 100 and Re = 400. In order to validate them, the results were compared to the benchmark values from [6], yielding Table C.1 and Table C.5. Furthermore, some simulations were done for Re = 1000, even though its convergence analysis was not done.

Moreover, in this case the use of a relaxation factor was considered so as to either enhance or not the performance of the code, in which it was seen that, even though it is not always optimal exactly at $f_r = 1.0$, its efficiency is rather high as Figure C.5 shows for Re = 100.

In order to enhance the performance of the code, in this point the Gauss-Seidel line-by-line method was replaced by the Conjugate Gradient Method (algorithm 3), one of the most known Krylov Subspace solvers, which converges much faster than the first used method even though it requires that the coefficients matrix A corresponds to a positive definite matrix. Even though some problems arose during the programming of the algorithm, its use has been proper since then, when the case simulated fulfills the properties required by the method.

The results obtained by using both methods for Re = 1000 and compared to the benchmark can be seen in Figure 3.4 and Figure 3.5 even though the convergence results can be seen in Appendix C.

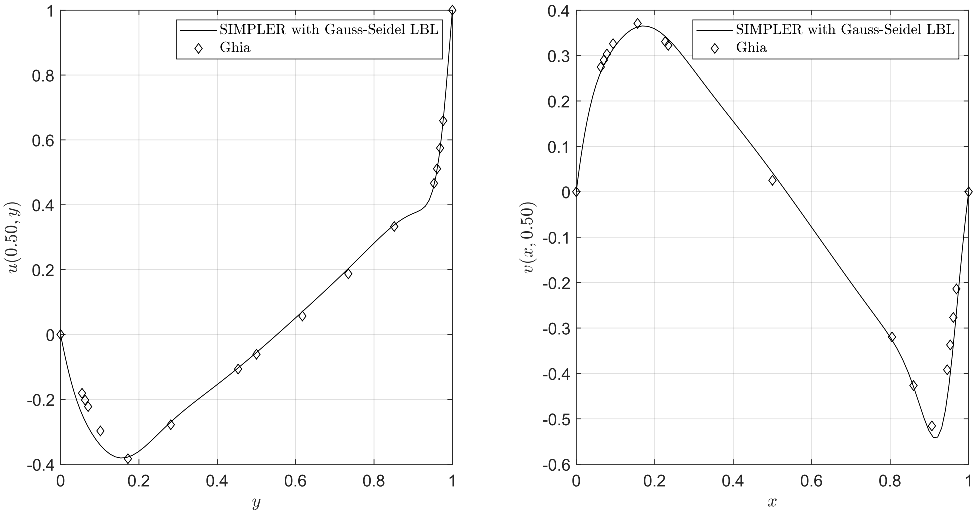


Figure 3.4: Comparison between the results at the specified points calculated using the SIMPLER algorithm using the LBL method and the results given in [6] using a 80×80 mesh.

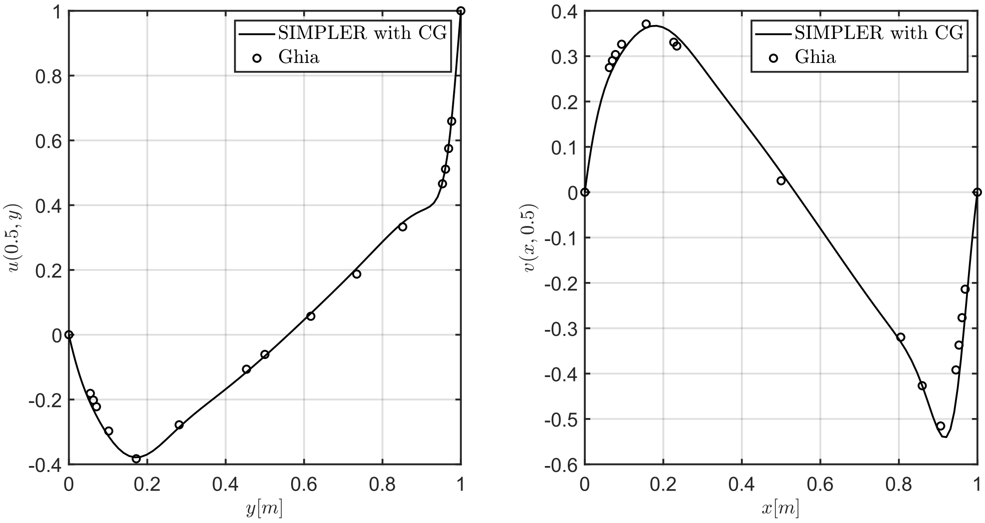


Figure 3.5: Comparison between the results at the specified points calculated using the SIMPLER algorithm using the CG algorithm and the results given in [6] using a 80×80 mesh.

Algorithm 2: Semi-Implicit Method for Pressure-Linked Equations Revised

```
$u_{i,g}(x, y) := u_{i,0}(x, y)$ ;
$t := 0$;
while $\varepsilon_t > \delta_t$ do
    $u_{i,P_{u_i},old} = u_{i,P_{u_i}}$ for $F = N, S, E, W$ do
        $a_{F_{u_i}} := D_f A(|\mathrm{Pe}_f|) + \max(-n_f F_f, 0)$;
    end
    $\hat{u}_{iP_u} = \frac{\sum_{\forall F} a_{F,i} u_{F,i} + b}{a_{P_{u_i}}}$;
    for $f = n, s, e, w$ do
        $d_f := \Delta_{x_f} n_f / a_{F_u}$;
    end
    for $F = N, S, E, W$ do
        $a_F = \rho_f d_f n_f \Delta_{x_f}$;
    end
    $b_P(\hat{u}_i) = \rho[\hat{u}_w - \hat{u}_e]\Delta_y + \rho[\hat{v}_s - \hat{v}_n]\Delta_x$;
    Solve $a_P p_P^* = \sum_{\forall F} a_F p_F^* + b_P$;
    for $i$=1:2 do
        Solve $a_{P_{u_i}} u_{P_{u_i}}^* = \sum_{\forall F_{u_i}} a_{F_{u_i}} u_{F_{u_i}}^* + b_{P_{u_i}} - \frac{\partial p^*}{\partial x_i}\Big|_{P_{u_i}}$;
    end
    $b_P(u_i^*)$;
    Solve $a_P p_P^{'=\sum_{\forall F} a_F p^{'F} + b_P}$;
    $u_{P_u} = u_{P_u}^* + d_e(p'_P - p'_E)$;
    $v_{P_v} = v_{P_v}^* + d_n(p'_P - p'_N)$;
    $\varepsilon_t = \frac{\max|u_{i,P_{u_i}} - u_{i,P_{u_i},old}|}{\Delta_t}$;
    $t := t + \Delta_t$;
end
```

Algorithm 3: Conjugate Gradient Method

```
$\mathbf{r_0} := \mathbf{b} - \mathbf{A}\mathbf{x_0}$;
if $\mathbf{r_0}$ is sufficiently small then
    $\mathbf{x} := \mathbf{x_0}$;
else
    $\mathbf{p_0} := \mathbf{r_0}$;
    $k := 0$;
    while $\mathbf{r}_k < \delta$ do
        $\alpha_k := \frac{\mathbf{r_k^T}\mathbf{r_k}}{\mathbf{p_k^T}\mathbf{A}\mathbf{p_k}}$;
        $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k\mathbf{p_k}$ ;
        $\mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{p_k}$ ;
        if $\mathbf{r}_{k+1} < \delta$ then
            Exit loop;
        else
            $\beta_k := \frac{\mathbf{r}_{k+1}^T\mathbf{r}_{k+1}^T}{\mathbf{r}_k^T\mathbf{r}_k^T}$;
            $\mathbf{p}_{k+1} := \mathbf{r}_{k+1} + \beta_k\mathbf{p}_k$;
            $k := k + 1$;
        end
    end
end
```

3.3.2 Fractional Step Method

The Fractional Step Method (FSM) (algorithm 4) is based in the Helmoltz-Hodge theorem, which states that a divergence-free vector field such as u_i in an incompressible flow will be orthogonal to the divergence of a field, being in this case $\partial_i p$. A more formal approach to the method is developed in chapter 6.

Thus, the method computes a predictor velocity out of the subspace in which the equations must remain included and then, due to the application of the pressure gradient, this predictor velocity is projected in the space. Thus, the FSM may also be known as Projection Method. (See [4] for further information).

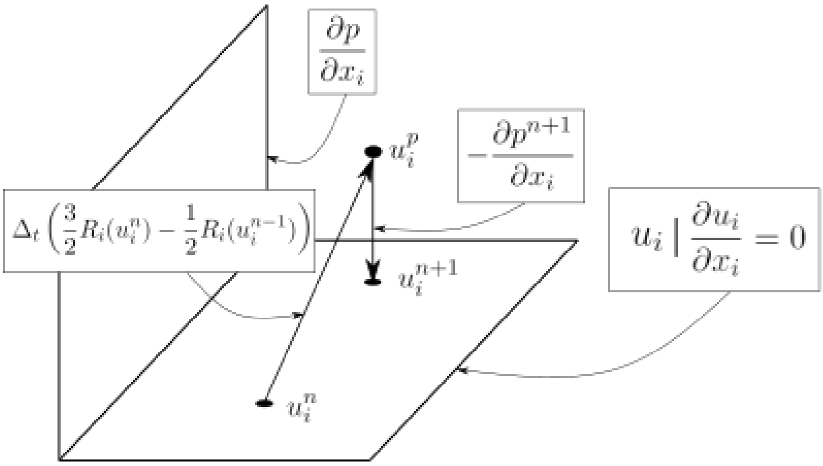

$\frac{\partial p}{\partial x_i}$
u_i^p
$-\frac{\partial p^{n+1}}{\partial x_i}$
$\Delta_t\left(\frac{3}{2}R_i(u_i^n) - \frac{1}{2}R_i(u_i^{n-1})\right)$
$u_i \mid \frac{\partial u_i}{\partial x_i} = 0$
u_i^{n+1}
u_i^n

Figure 3.6: Outline of how the Fractional Step Method works in terms of subspaces.

The predictor velocity is obtained by removing the pressure gradient of the momentum equation, which is discretized using a 2nd order Adams-Bashforth scheme.

Given that it corresponds to a implicit method, its stability will strongly rely on the timestep taken, which will have to be adapted to the problem using the CFL condition (3.7).

$$\Delta_t = \min\left[\min\left(0.35\frac{\Delta_x}{|\mathbf{u}|}\right), \min\left(0.20\frac{\rho\Delta_x^2}{\mu}\right)\right] \tag{3.7}$$

By using this method, two different cases were solved. First of all, the LDC as done using the SIMPLER, from Reynolds numbers going from 100 to 3200 and available at [5]. In this case there is no convergence plot, even though its convergence was also analyzed, providing a suitable mesh for 100, 400 and 1000 at around 10000 control volumes, whereas for the latter 3200, a converged mesh was not found due to lack of computational power in the laptop used then. These results can be seen in Appendix C.

On the other hand, a DHC, which is described in Appendix D, was also programmed, by adding the energy equation on the set solved. This addition modified a little bit algorithm 4, since a flux term for temperature (3.8) as well as its calculation (3.9) using the same numerical scheme were added.

$$R_T = -\frac{\lambda}{\rho C_p}\frac{\partial^2 T}{\partial x_j \partial x_j} - u_j\frac{\partial T}{\partial x_j} \tag{3.8}$$

$$T^{n+1} = T^n + \Delta_t\left(\frac{3}{2}R_T^n - \frac{1}{2}R_T^{n-1}\right) \tag{3.9}$$

Moreover, the Boussinesq approximation (2.4) had to be included in the momentum equation so as to consider the changes of density due to temperature, adding a body forces term

$$\rho(T)g_i \tag{3.10}$$

Algorithm 4: Fractional Step Method

```
$t := 0$ ;
$n := 1$ ;
for $F = N, S, E, W$ do
    $a_F := \frac{|n_f|\Delta_{x_f}}{\Delta_{PF}}$;
end
$a_P := \sum_{\forall F} a_F$;
while $\varepsilon_t > \delta_t$ do
    $R_u = \frac{1}{\Omega_x}\left[\rho\left(\Delta_y(u_w^2 - u_e^2) + \Delta_x(u_s v_s - u_n v_n)\right) + \frac{\mu_e \Delta_y}{\Delta_x^{PE}} u_{E_u} + \frac{\mu_w \Delta_y}{\Delta_x^{PW}} u_{W_u} + \right.$
    $\left. + \frac{\mu_n \Delta_x}{\Delta_y^{PN}} u_{N_u} + \frac{\mu_s \Delta_x}{\Delta_y^{PS}} u_{S_u} - \left(\Delta_y\left(\frac{\mu_e}{\Delta_x^{PE}} + \frac{\mu_w}{\Delta_x^{PW}}\right) + \Delta_x\left(\frac{\mu_n}{\Delta_y^{PN}} + \frac{\mu_s}{\Delta_y^{PS}}\right)\right) u_{P_u}\right]$;
    $R_v = \frac{1}{\Omega_y}\left[\rho\left(\Delta_y(u_w v_w - u_e v_e^2) + \Delta_x(v_s^2 - v_n^2)\right) + \frac{\mu_e \Delta_y}{\Delta_x^{PE}} v_{E_v} + \frac{\mu_w \Delta_y}{\Delta_x^{PW}} v_{W_v} + \right.$
    $\left. + \frac{\mu_n \Delta_x}{\Delta_y^{PN}} v_{N_v} + \frac{\mu_s \Delta_x}{\Delta_y^{PS}} v_{S_v} - \left(\Delta_y\left(\frac{\mu_e}{\Delta_x^{PE}} + \frac{\mu_w}{\Delta_x^{PW}}\right) + \Delta_x\left(\frac{\mu_n}{\Delta_y^{PN}} + \frac{\mu_s}{\Delta_y^{PS}}\right)\right) v_{P_v}\right]$;
    $u_{i,P_{u_i}}^p = u_P^n + \frac{\Delta_t}{\rho}\left(\frac{3}{2}R_i^n - \frac{1}{2}R_i^{n-1}\right)$;
    $b_P = -\frac{\rho}{\Delta_t}\left[(u_{P_u}^p - u_{W_u}^p)\Delta_y + (v_{P_v}^p - v_{S_v}^p)\Delta_x\right]$;
    Solve $a_P p_P = a_E p_E + a_W p_W + a_N p_N + a_S p_S + b_P$ ;
    $u_{i,P_{u_i}}^{n+1} = u_{i,P_{u_i}}^p - \frac{\Delta_t}{\rho} \left.\frac{\partial p}{\partial x_i}\right|_{P_{u_i}}$;
    $t := t + \Delta_t$;
    $n := n + 1$;
end
```

Thus, the case solved, based on [7], was for Pr = 0.71 and Ra = $10^3, 10^4, 10^5, 10^6$, where Pr is the Prandtl number (3.11) and Ra is the Rayleigh number (3.13).

$$\mathrm{Pr} = \frac{\nu}{\alpha} \tag{3.11}$$

where α stands for the thermal diffusivity (3.12)

$$\alpha = \frac{\lambda}{C_p \rho} \tag{3.12}$$

$$\mathrm{Ra} = \frac{\rho^3 C_p \beta D^3 g \Delta T}{\mu \lambda} \tag{3.13}$$

After implementing the code, the results were compared to De Vahl Davis' benchmark [8], the convergence to which was not extremely good even though the convergence of the code was as expected, as can be seen in Figure 3.7.

Moreover, an analysis of the computational cost as a function of the number of control volumes as well as the Rayleigh was performed, obtaining its dependance follows $t_{COMP} \sim \mathcal{O}(\mathrm{Ra}^{0.1717})$.

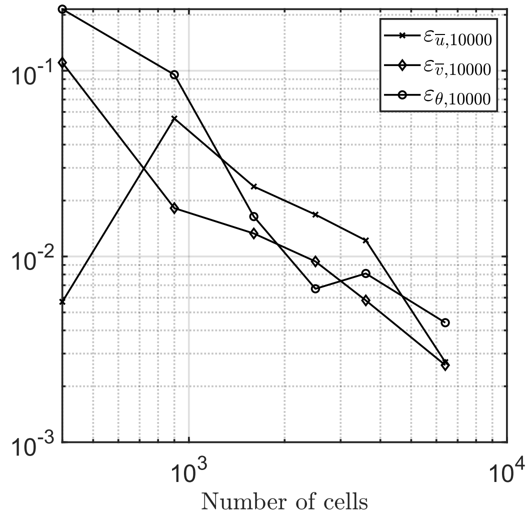


Figure 3.7: Convergence plots for velocities and temperature at Ra $= 10^3$. Extracted from [9].

Chapter 4

Turbulence. Theory

4.1 Introduction to turbulent flows

A basic methodology to classify all sorts of flows corresponds to distinguishing by regime, where there are two global possibilities: either corresponds to a laminar or a turbulent flow.

Laminar flows follow a deterministic behaviour: if an experiment is done under these circumstances (characterized generally by Re) the result will be the same, without considering possible experimental errors, in each and every case, following the classical physics determinism (as expected given that NS are deterministic equations).

As it can be expected, nonetheless, most flows in nature are turbulent, since the required conditions for having a laminar flow are seldom available in nature, given that even a small perturbation can generate a turbulent flow, even at low Re.

Kundu [10] states, nonetheless, five different common characteristics for turbulent flows:

- Randomness: flows in turbulent regime are chaotic and generally unpredictable. Thus, statistical approaches have to be performed.
- Nonlinearity: turbulent flows, as well as laminar, are nonlinear in the sense that convection is highly nonlinear, provided that the velocity structures transport themselves. Nonetheless, in the case of turbulent flows, its three-dimensional characteristics induces a vortex stretching process.
- Diffusivity: due to a high mixing of fluid particles, there is a rapid rate of momentum diffusion.
- Vorticity: a main characteristic of turbulent flows is the appearance of vorticity structures (called eddies), whose size range is very wide, going from macroscopical eddies (e.g. the size of the boundary layer) to microscopical eddies, with the energy flowing from higher to smaller scales, known as *energy cascade.*
- Dissipation: when energy reaches the smallest scales, it is dissipated by the viscosity.

4.2 Mean-flow equations

4.2.1 Reynolds equations

Historically, what has been done to deal with the randomness of the turbulent flows is time-averaging the equations. In order to do so, the time average of a function V will be defined as $\langle V\rangle$ in (4.1), where $f(V)$ is the probability density function (PDF) for V

$$\langle V\rangle = \int_{-\infty}^{+\infty} V f(V)\, dV \tag{4.1}$$

The main properties of the mean value used will be summarized, being U and V two independent functions and a and b two real known parameters, as

$$\langle aU + bV\rangle = a\langle U\rangle + b\langle V\rangle$$

Then, for the case of the velocity field, Reynolds stated that it could be interpreted as (4.2), being the sum of the mean value and its perturbations.

$$u_i = \langle u_i\rangle + u_i' \tag{4.2}$$

Thus, the time average of NS has to be done. Nevertheless, all derivations will be done considering an incompressible flow for the sake of simplicity.

For the case of continuity equation (2.2) it follows that both mean and perturbations will have a null divergence (4.3).

$$\frac{\partial u_i'}{\partial x_i} = 0 \tag{4.3a}$$

$$\frac{\partial \langle u_i\rangle}{\partial x_i} = 0 \tag{4.3b}$$

In taking the mean of the momentum equation (2.3), the mean of each term has to be considered. Even though in most of the cases it is trivial, that does not apply in the convective term.

Converting it onto the conservative form and taking the average value, (4.4) arises.

$$\frac{\partial}{\partial x_j}\langle u_i u_j\rangle = \frac{\partial}{\partial x_j}\langle(\langle u_i\rangle + u_i')(\langle u_j\rangle + u_j')\rangle = \frac{\partial}{\partial x_j}\langle\langle u_i\rangle\langle u_j\rangle + u_j'\langle u_i\rangle + u_i'\langle u_j\rangle + u_i'u_j'\rangle \tag{4.4}$$

Given that the mean value of the perturbations is zero, as shown in section 4.2.1 and that $\langle u_j'\langle u_i\rangle\rangle = \langle u_j'\rangle\langle u_i\rangle = 0$, (4.4) yields (4.5).

$$\langle u_i u_j\rangle = \langle u_i\rangle\langle u_j\rangle + \langle u_i'u_j'\rangle \tag{4.5}$$

Thus, the time averaged convective term corresponds to (4.6).

$$\left\langle \frac{Du_i}{Dt}\right\rangle = \frac{\partial\langle u_i\rangle}{\partial t} + \langle u_j\rangle\frac{\partial\langle u_i\rangle}{\partial x_j} + \frac{\partial}{\partial x_i}\langle u_i'u_j'\rangle \tag{4.6}$$

Now, by defining the mean substantial derivative (4.7), as stated by [2], the Reynolds equations can be found (4.8), which follow a similar structure to NS, yet with the addition of an extra term, called as Reynolds stresses.

$$\frac{\bar{D}}{\bar{D}t} = \frac{\partial}{\partial t} + \langle u_j\rangle\frac{\partial}{\partial x_j} \tag{4.7}$$

$$\rho\frac{\bar{D}\langle u_i\rangle}{\bar{D}t} = \mu\frac{\partial^2\langle u_i\rangle}{\partial x_j\partial x_j} - \rho\frac{\partial\langle u_i'u_j'\rangle}{\partial x_i} - \frac{\partial\langle p\rangle}{\partial x_i} \tag{4.8}$$

Proof of the null mean value of the perturbations

Given (4.2):

$$\langle u_i \rangle = \langle \langle u_i \rangle + u_i' \rangle = \langle u_i \rangle + \langle u_i' \rangle \tag{4.9}$$

Thus, $\langle u_i' \rangle = 0$

4.2.2 Reynolds stresses

Previously the term $\langle u_i' u_j' \rangle$ has been defined as Reynolds stresses.

In order to see this stress meaning, (4.8) can be rewritten as a general momentum conservation equation (4.10), where the Reynolds stress now gets its clear meaning, given the similarity in definition to viscous and pressure stresses.

$$\rho \frac{\bar{D}\langle u_i \rangle}{\bar{D}t} = \frac{\partial}{\partial x_i}\left[\mu\left(\frac{\partial \langle u_i \rangle}{\partial x_j} + \frac{\partial \langle u_j \rangle}{\partial x_i}\right) - \langle p \rangle \delta_{ij} - \rho \langle u_i' u_j' \rangle\right] \tag{4.10}$$

Thus, the actual stress will be, in fact, $-\rho\langle u_i' u_j' \rangle$ yet by the definition in (4.8) it is more natural to define the Reynolds stress as previously detailed.

This, however, implies that more variables appear in the problem: the velocity perturbations, which was what was intended to avoid. Thus, the problem is not closed unless the Reynolds stresses are determined, being this where the models appear.

At this point, the turbulent kinetic energy will be defined as (4.11).

$$k = \frac{1}{2}\mathrm{tr}(\langle u_i' u_j' \rangle) \tag{4.11}$$

Nevertheless, now the Reynolds stress tensor can be separated into its isotropic and its deviatoric part (also called anisotropic). In order to find the latter, a_{ij}, (4.12) will be applied. In fact, the turbulent kinetic energy will be removed yielding just cross terms.

$$a_{ij} = \langle u_i' u_j' \rangle - \frac{2}{3}k\delta_{ij} \tag{4.12}$$

4.2.3 Mean scalar equation

The convection-diffusion equation (4.13), which stands for a conservation equation for different scalars (e.g. temperature) can also be time averaged.

$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial x_j}(u_j \phi) = \Gamma \frac{\partial^2 \phi}{\partial x_j \partial x_j} \tag{4.13}$$

Thus, applying (4.2) and its equivalent to the scalar (4.14) to (4.13) using an equivalent approach to the Reynolds equations, (4.15) is obtained.

$$\phi = \langle \phi \rangle + \phi' \tag{4.14}$$

$$\frac{\partial \langle \phi \rangle}{\partial t} + \langle u_j \rangle \frac{\partial \langle \phi \rangle}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\Gamma \frac{\partial \langle \phi \rangle}{\partial x_j} - \langle u_j' \phi' \rangle\right) \tag{4.15}$$

This, nevertheless, will not close the problem since it adds two more variables related to the scalar: its mean value and perturbations, which appear in the scalar flux ($\langle u_j' \phi' \rangle$).

4.2.4 Closure of the problem

As seen in both (4.8) and (4.15), the problem requires a model for both scalar flux and Reynolds stresses. These are known as the gradient-diffusion and turbulent-viscosity hypotheses, respectively.

Turbulent-viscosity hypothesis

The turbulent-viscosity hypothesis indicates that the deviatoric Reynolds stress (a_{ij}) will be proportional to the mean rate of strain $\overline{S}_{ij}$. Thus, (4.16) will be found, at which the turbulent viscosity is defined:

$$-\rho\langle u_i'u_j'\rangle + \frac{2}{3}\rho k\delta_{ij} = \rho\nu_T\left(\frac{\partial\langle u_i\rangle}{\partial x_j} + \frac{\partial\langle u_j\rangle}{\partial x_i}\right) \tag{4.16}$$

Gradient-diffusion hypothesis

The gradient-diffusion hypothesis indicates that the scalar flux can be interpreted as a turbulent transport of the scalar ϕ from larger to smaller values. Thus, it can be mathematically interpreted as (4.17), in which the turbulent diffusivity of the scalar is defined.

$$\langle u_i\phi'\rangle = -\Gamma_T\nabla\langle\phi\rangle \tag{4.17}$$

4.3 Classical theory of turbulence

4.3.1 The energy cascade

In 1922, Lewis Richardson [11] stated

Big whorls have little whorls,
Which feed on their velocity;
And little whorls have lesser whorls,
And so on to viscosity.

In fact, these lines might seem simple, yet the description of the energy cascade phenomena is clearly described. In fact, large eddies (or whorls, as stated by Richardson), with a lengthscale similar to $\mathcal{L}$, whereas its characteristic velocity does not differ much from the reference velocity $\mathcal{U}$; are characterized by a large unstability, leading them to breaking up and transferring its energy onto smaller eddies. These, at the same time, go through the same exact process until the Reynolds number on the eddies is sufficiently small that viscosity can dissipate the energy.

Thus, energy will be dissipated after a sequence of processes, which will be very important to determine the rate of dissipation ε. In fact, it will be determined by the energy transfer of the largest eddies, which are determined by a energy order of u_0^2 and a timescale of $\tau_0 = l_0/u_0$.

Thus, ε will be scaled as $u_0^2/\tau \equiv u_0^3/l_0$, being independent of ν, provided that only very high Re cases are considered.

4.3.2 Kolmogorov hypotheses

After presenting the energy cascade, Andrey Kolmogorov intended to provide an answer for different fundamental questions that arise from the theory.

Kolmogorov's hypothesis of local isotropy

Concerning the size corresponding to the eddies at which viscous dissipation appears, Kolmogorov [12] stated that *At sufficiently high Reynolds number, the small-scale turbulent motions are statistically isotropic.*, which is known as the local isotropy hypothesis.

As a difference to large eddies, which are noticeably affected by the boundary conditions, this hypothesis implies that, for $l < l_{EI}$, the eddies will not be affected by external conditions, being this the reason why the small-scale motions are usually said to be universal.

Kolmogorov's first similarity hypothesis

This leads onto Kolmogorov's first similarity hypothesis, which stated that every trubulent flow with high enough Reynolds number the small-scale, defined at $l < l_{EI}$ behaviour will be statistically universal and determined by ν and ε.

Thus, the small-scale range will be defined as the universal equilibrium range. This range will, in fact, have its own length,velocity and time scales, called Kolmogorov scales (4.18).

$$\eta = (\nu^3/\varepsilon)^{1/4} \tag{4.18a}$$

$$u_\eta = (\varepsilon\nu)^{1/4} \tag{4.18b}$$

$$\tau_\eta = (\nu/\varepsilon)^{1/2} \tag{4.18c}$$

These scales are determined to characterize the dissipative eddies, which are determined to have a Re = 1. Thus, from Reynolds number definition,

$$\nu = \eta u_\eta$$

as well as the definition of the rate of dissipation

$$\varepsilon = \frac{u_0^3}{l_0} = \frac{u_\eta^3}{\eta}$$

the relationships between large and small scales as a function of Re can be found.

By the definition of ε,

$$\left(\frac{u_0}{u_\eta}\right)^3 = \frac{l_0}{\eta}$$

Thus, assuming $\nu = \eta u_\eta$,

$$\mathrm{Re} = \frac{u_0 l_0}{\nu} = \frac{u_0 l_0}{u_\eta \eta} = \left(\frac{u_0}{u_\eta}\right)^4$$

Therefore,

$$\frac{u_\eta}{u_0} = \mathrm{Re}^{-1/4} \tag{4.19}$$

By applying it onto Re definition,

$$\mathrm{Re} = \mathrm{Re}^{1/4}\frac{l_0}{\eta}$$

Hence,

$$\frac{\eta}{l_0} = \mathrm{Re}^{-3/4} \tag{4.20}$$

Eventually, by applying the definition of time scales,

$$\frac{\tau_\eta}{\tau_0} = \frac{\eta/u_\eta}{l_0/u_0} = \frac{\eta}{l_0}\frac{u_0}{u_\eta}$$

Then,

$$\frac{\tau_\eta}{\tau_0} = \mathrm{Re}^{-1/2} \tag{4.21}$$

Kolmogorov's second similarity hypothesis

Nevertheless, at most turbulent conditions, there will be a set of lengthscales $l_0 >> l >> \eta$ such that its behaviour will be statistically universal, as previously defined by Kolmogorov's first similarity hypothesis, yet there will not be any kind of dissipation. Thus, this hypothesis states that "in every turbulent flow at sufficiently high Reynolds number, the statistics of motion of scale l in the range $l_0 >> l >> \eta$ have a universal form that is uniquely determined by ε, independent of ν" [2].

Thus, this range will be called inertial subrange that will be limited by l_{EI}, which accounts for the start of the universal equilibrium range, and l_{DI}, which determines the start of the dissipation range. This can be summarized by Figure 4.1.

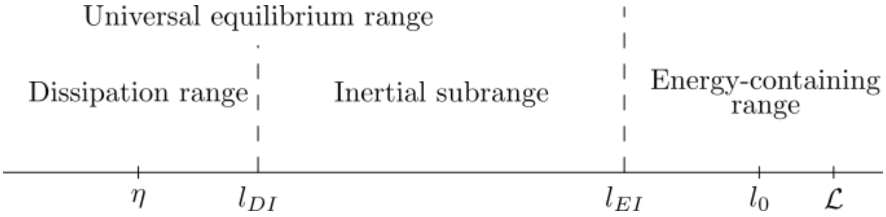


Figure 4.1: Outline for the turbulent behaviour for different lengthscales. Adapted from [2].

Thus, the energy cascade will clearly be explained by the production of energy in the energy containing range, which at the same time will be transported inertially through the inertial subrange and finally will be dissipated at the dissipation range, which can be outlined by Figure 4.2.

4.3.3 Energy spectrum

Starting from a dimensional analysis of the energy spectrum, which corresponds to $\mathrm{J \cdot m \cdot kg^{-1}}$, yielding L^3T^{-2} and provided that the energy spectrum in the inertial range will just depend on κ (L^{-1}) and ε ($\mathrm{L}^2\mathrm{T}^{-3}$), it can be derived that the energy spectrum will be proportional to $\varepsilon^{2/3}\kappa^{-5/3}$. Thus, the compensated Kolmogorov spectrum function ($\Psi(\kappa\eta)$) will act as the proportionallity parameter, yielding (4.22).

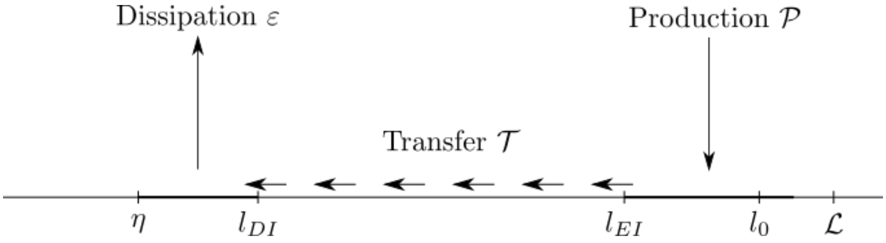


Figure 4.2: Outline for the energy cascade at turbulent regime. Adapted from [2].

$$E(\kappa) = \varepsilon^{2/3}\kappa^{-5/3}\Psi(\kappa\eta) \tag{4.22}$$

Nonetheless, in the inertial range $l_{DI} < l < l_{EI}$, which converted to wavenumbers will correspond to $\kappa_{DI} > \kappa > \kappa_{EI}$. Thus, the relation will still be valid if multiplied by a constant, which is in this case η provided that the parameter of interest for Ψ is $\kappa\eta$. From here it follows that $\kappa_{DI}\eta > \kappa\eta > \kappa_{EI}\eta$.

Since $\mathcal{O}(l_{DI}) \approx \mathcal{O}(\eta)$, $\kappa_{DI}\eta \approx 1$, $\kappa \ll 1$. Thus, Ψ can be considered as a constant, which will be replaced by $_K$, yielding (4.23).

$$E(\kappa) = C_K\varepsilon^{2/3}\kappa^{-5/3} \tag{4.23}$$

This value $p = -\frac{5}{3}$ can also be obtained by considering the energy spectrum as a general power-law spectrum

$$E = A\kappa^{-p}$$

Thus, following the definitions of energy in a wavenumber range (4.24)

$$k_{(\kappa_a,\kappa_b)} = \int_{\kappa_a}^{\kappa_b} E(\kappa)\ d\kappa \tag{4.24}$$

and the definition (see Section 6.5 from [2]) of the dissipation rate in a wavenumber range (4.25)

$$\varepsilon_{(\kappa_a,\kappa_b)} = \int_{\kappa_a}^{\kappa_b} 2\nu\kappa^2 E(\kappa)\ d\kappa \tag{4.25}$$

it can be seen that the energy contained in all wavenumbers higher than κ corresponds to (4.26)

$$k_{(\kappa,\infty)} = \int_{\kappa}^{\infty} A\kappa^{-p}\ d\kappa = \frac{A}{p-1}\kappa^{-(p-1)} \tag{4.26}$$

in which the energetic behaviour will be shown. For $p < 1$, it can be shown that the energy level will be bounded, whereas for $p \geq 1$ the energy level will not be bounded for $\kappa \to \infty$.

On the other hand, the dissipation rate in all wavenumbers lower than κ corresponds to (4.27).

$$\varepsilon_{(0,\kappa)} = \int_{0}^{\kappa} 2\nu\kappa^2 A\kappa^{-p}\ d\kappa \tag{4.27}$$

In this case, for $p < 3$, the dissipation rate will be higher for higher κ, as follows from Kolmogorov's second similarity hypothesis, whereas for $p = 3$ the dissipation rate diverges and for $p > 3$ the dissipation rate does not follow the behaviour from Kolmogorov's hypotheses.

Thus, Kolmogorov spectrum is defined by $p = \frac{5}{3}$ provided that it corresponds to the mid value of the interval $(1, 3)$, in which both dissipation rate and energy contained are physically and mathematically acceptable.

4.4 Burgers' equation

A simple way to see how turbulence behaves is using the Burgers' equation , which stands for a unidimensional momentum conservation-like equation without considering the pressure gradient and, instead, considering a source term f, which yield (4.28).

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}=\frac{1}{\mathrm{Re}}\frac{\partial^2 u}{\partial x^2}+f \tag{4.28}$$

Thus, as Kolmogorov's turbulent theory proposes, the velocity field u can be transformed onto the Fourier space as (4.29) indicates, where $\hat{u}_k = f(t)$ while the space dependance will rely on the trigonometric part.

$$u=\sum_{k=-N}^{k=+N}\hat{u}_k e^{ikx} \tag{4.29}$$

From this statement, as well as having f modified using an equivalent method as (4.29), (4.28) yields (4.30), which will stand for all wavenumbers k desired.

$$\partial_t \hat{u}_k+\sum_{p+q=k}\hat{u}_p iq\hat{u}_q=-\frac{k^2}{\mathrm{Re}}\hat{u}_k+\hat{F}_k \tag{4.30}$$

Nonetheless, the turbulent theory does not consider velocity fields yet energy spectra, which, indeed, will be found for each wavenumber just by using (4.31).

$$E_k=\hat{u}_k\overline{\hat{u}}_k \tag{4.31}$$

Thus, this equation is easily solvable for a DNS, which allows to see the true solution for the desired case, which was set for Re = 40, provided that it would generate results similar to Kolmogorov's theory.

In order to do so, an Adams-Bashforth second order scheme was used to discretize in time, whereas a space discretization was not required provided that $\hat{u}_k$ does not depend on x.

On the other hand, the convective term was calculated for all positive terms from the equation $p+q=k$, considering that $\hat{u}_k=\overline{\hat{u}_{-k}}$, where $\overline{(\cdot)}$ stands for the complex conjugate.

Thus, Figure 4.3 is obtained, where the energy cascade can be clearly observed.

On the other hand, Figure 4.4 shows a case run for $k_N = 500$ and Re = 100, in which the energy level at higher wavenumbers is higher, yet, in the end, it gets dissipated at the highest wavenumbers.

In fact, the Burgers' equation is also an excellent tool to test LES models, which will be seen in chapter 5, where Burgers' equation is again used to illustrate the models.

4.5 Laminar boundary layers

Consider a plane flow with a certain viscosity ν and a free stream velocity $\mathcal{U}$ past a slender cylindrical body. Due to the no-slip condition at the wall ($u_i = 0$), the velocity component tangent to the body will have a gradient in the normal direction from 0 to $u_i = u_i(x)$, known as boundary layer. In this region, characterized by its height δ, the velocity gradient will be

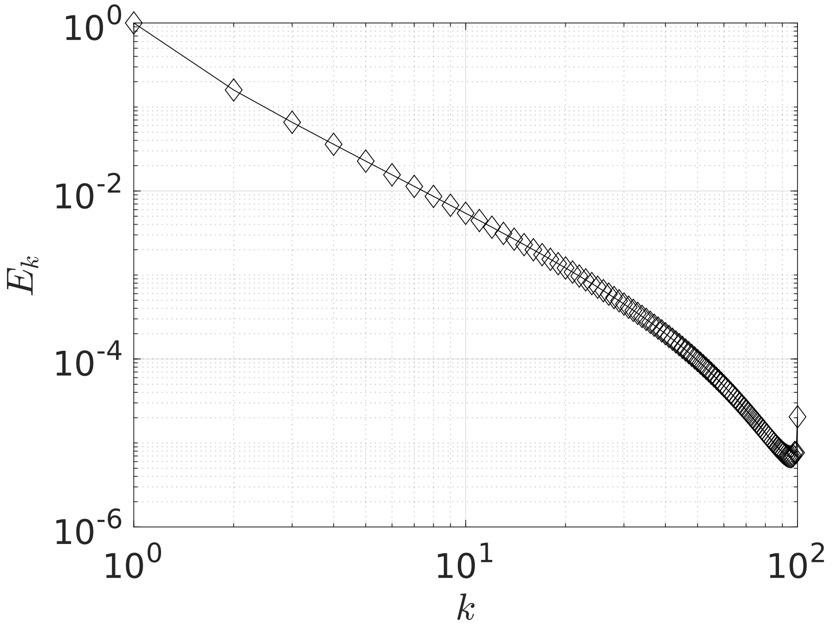


Figure 4.3: DNS of the Burgers' equation with Re = 40 using a second order Adams-Bashforth scheme.

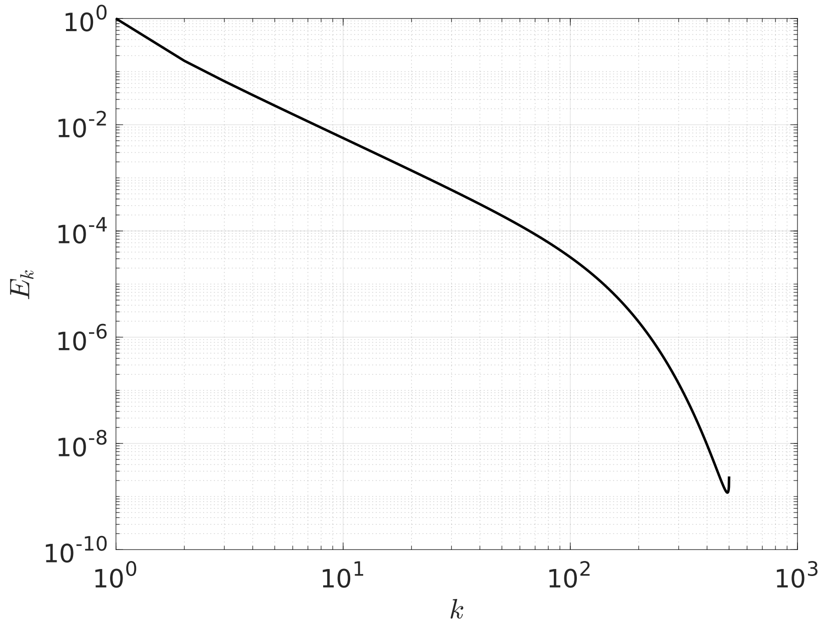


Figure 4.4: DNS of the Burgers' equation with Re = 100 using a second order Adams-Bashforth scheme.

very large, being this the reason why the viscosity will have a big implication in the shear stress, defined by (4.32).

$$\tau = \mu \frac{\partial u_i}{\partial \hat{n}} \tag{4.32}$$

Nonetheless, in the outer region to the boundary layer there will not be large viscosity gradient, being this the reason why the viscosity will not have such an important role. The flow can be considered frictionless and potential.

This phenomena is illustrated in Figure 4.5, where the dashed line represents the local height of the boundary layer.

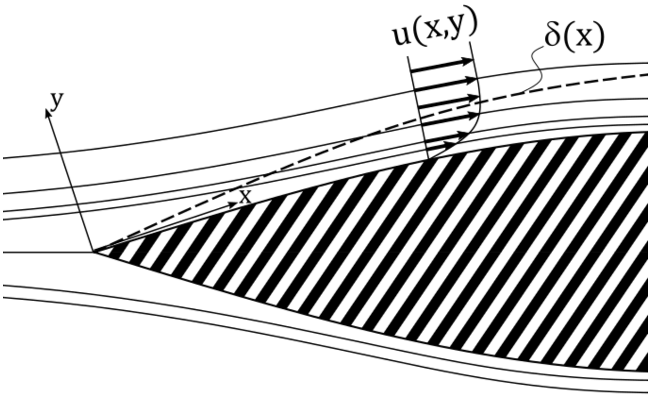


Figure 4.5: Boundary layer flow along a wall. Adapted from [13].

4.5.1 Prandtl Boundary-Layer Equations

From (2.8), the orders of magnitude of each member of the equations can be analyzed. In order to do so, Schlichting and Gersten [13] state that $\mathcal{O}(\hat{u}) = \mathcal{O}(\hat{x}) = \mathcal{O}(\hat{t}) = 1$, whereas $\mathcal{O}(\hat{v}) = \mathcal{O}(\hat{y}) = \hat{\delta}$. On the other hand, analytical solutions from (2.3) show that $\delta \sim \sqrt{\nu}$ (see Chapter 5 from [13]). Thus, $\mathcal{O}(\text{Re}) = \frac{1}{\hat{\delta}^2}$. Thus, in the momentum equation in the x direction the order of magnitude will correspond to (4.33).

$$\mathcal{O}\left(\frac{\partial \hat{u}}{\partial \hat{t}} + \hat{u}\frac{\partial \hat{u}}{\partial \hat{x}} + \hat{v}\frac{\partial \hat{u}}{\partial \hat{y}}\right) = \mathcal{O}\left[-\frac{\partial \hat{p}}{\partial \hat{x}} + \frac{1}{\text{Re}}\left(\frac{\partial^2 \hat{u}}{\partial \hat{x}^2} + \frac{\partial^2 \hat{u}}{\partial \hat{y}^2}\right)\right]$$

$$1 + 1\frac{1}{1} + \hat{\delta}\frac{1}{\hat{\delta}} = \mathcal{O}\left(-\frac{\partial \hat{p}}{\partial \hat{x}}\right) + \hat{\delta}^2\left(1 + \frac{1}{\hat{\delta}^2}\right) \tag{4.33}$$

Moreover, $\hat{y}$ will have to assume very small values for the boundary layer, it will not be suitable for it to describe it. Therefore, $\hat{v}$ will also not be suitable for the same reason. From this idea follows the boundary-layer transformation (4.34).

$$\overline{(\cdot)} = \hat{(\cdot)}\sqrt{\text{Re}} \tag{4.34}$$

By doing the exact same procedure for the momentum equation in y direction, as well as for the continuity equation, the Prandtl boundary-layer equations (4.35) are derived.

$$\frac{\partial \hat{u}}{\partial \hat{x}} + \frac{\partial \overline{v}}{\partial \overline{y}} = 0 \tag{4.35a}$$

$$\frac{\partial \hat{u}}{\partial \hat{t}} + \hat{u}\frac{\partial \hat{u}}{\partial \hat{x}} + \overline{v}\frac{\partial \hat{u}}{\partial \overline{y}} = -\frac{\partial \hat{p}}{\partial \hat{x}} + \frac{\partial^2 \hat{u}}{\partial \overline{y}^2} \tag{4.35b}$$

$$0 = -\frac{\partial \hat{p}}{\partial \overline{y}} \tag{4.35c}$$

4.5.2 Wall friction

In this thesis, the aim behind solving and understanding how boundary layers behave is obtaining both lift and drag coefficients. Even though some fraction of these forces is generated by the pressure gradient, most important at the lift, an important part of the drag comes from the wall friction. Thus, it is very important to compute this magnitude τ_w. In fact, provided that the whole development has been done dimensionless, the skin-friction coefficient c_f will be defined by (4.36), in which the wall friction is scaled by the dynamic pressure $\frac{1}{2}\rho\mathcal{U}^2$.

$$c_f(\hat{x}) = \frac{\tau_w(\hat{x})}{\frac{\rho}{2}\mathcal{U}^2} = \frac{2\mu}{\rho\mathcal{U}^2}\left(\frac{\partial u}{\partial y}\right)_w = \cdots = \frac{2}{\sqrt{\mathrm{Re}}}\left(\frac{\partial \hat{u}}{\partial \overline{y}}\right)_w \tag{4.36}$$

Thus, it can be seen that, for all laminar boundary layers, the skin-friction coefficient tends to zero for Re → ∞

4.5.3 Separation point

A particular case is where $c_f(\hat{x}) = 0$, called separation point, which occures with adverse pressure gradient. Thus, the flow cannot move onto the regions with high pressure due to its lower kinetic energy - in the potential zone, an adverse pressure gradient just implies a reduction of velocity - and moves onto lower pressure zones, what implies that the flow has now separated from the body onto the main potential flow.

4.6 Turbulent boundary layers

In order to understand and develop the theory for turbulent boundary layers, a turbulent Couette flow will be studied. The Couette flow is defined by a fixed plate separated $2H$ from a plate moving at a fixed velocity u_{wu}. For a fully developed flow and constant physical properties, a constant, time-averaged, shear force $\langle\tau_w\rangle$ has to be applied to the upper plate in order to keep the motion at u_{wu}.

Thus, the balance of forces (4.38) implies that this shear force has to be kept constant through all the flow.

Considering a time-averaged steady state flow, the mean equations yield, in the x direction, (4.37)

$$\rho\left(\langle u\rangle\frac{\partial\langle u\rangle}{\partial x} + \langle v\rangle\frac{\partial\langle u\rangle}{\partial y}\right) = -\frac{\partial\langle p_e\rangle}{\partial x} + \frac{\partial}{\partial_y}(\langle\tau_w\rangle + \tau_t) \tag{4.37}$$

Since it is a fully developed flow, $\partial\langle u\rangle/\partial x = 0$ and $\langle v\rangle = 0$, and there is no external pressure applied, it is a pure shear flow [13]. Thus,

$$\frac{\partial}{\partial y}(\langle \tau_w \rangle + \tau_t) = 0$$

what yields (4.38)

$$\langle \tau_w \rangle = \langle \tau_\nu \rangle + \tau_t = ct. \tag{4.38}$$

where:

$\langle \tau_\nu \rangle = \rho\nu\frac{d\langle u\rangle}{dy}$

$\tau_t = -\rho\langle u'v'\rangle$

Thus, there are just two methods for transferring momentum: the molecular viscous transfer from $\langle \tau_\nu \rangle$ and the turbulent momentum transfer from τ_t.

4.6.1 Wall units

By expanding $\langle \tau_w \rangle$, it can be seen that only $\langle \tau_w \rangle/\rho$ appears. Thus, the velocity field must depend on $\langle u\rangle = f(y, H, \nu, \langle \tau_w \rangle/\rho)$. Following the Π theorem, a new quantity called friction velocity (or wall friction velocity) u_τ (4.39) might be introduced, being "the characteristic velocity for turbulent flows given a wall shear stress [13]".

$$u_\tau = \sqrt{\frac{\langle \tau_w \rangle}{\rho}} \tag{4.39}$$

Thus, dimensionless parameters for the boundary layers follow this definition:

$\eta = \frac{y}{H}$

$u^+ = \frac{\langle u\rangle}{u_\tau}$

$\mathrm{Re}_\tau = \frac{u_\tau H}{\nu}$

$\tau_t^+ = \frac{\tau_t}{\rho u_\tau^2}$

Applying these parameters, $u^+ = f(\eta, \mathrm{Re}_\tau)$. Furthermore, this can also be applied at (4.38), yielding the differential equation from (4.40).

$$u_\tau^2\rho = \rho\nu\frac{\partial(u^+u_\tau)}{\partial(H\eta)} + \tau_t$$

$$1 = \frac{\nu u_\tau}{u_\tau^2 H}\frac{\partial u^+}{\partial\eta} + \tau_t^+$$

$$\frac{1}{\mathrm{Re}_\tau}\frac{\partial u^+}{\partial\eta} + \tau_t^+ = 1 \tag{4.40}$$

which have to follow at $\eta = 0$, $u^+ = \tau_t^+ = 0$ and at $\eta = 1$, $du^+/d\eta^2 = 0$. Nonetheless, it is an open problem given that there are two unknowns u^+, τ_t^+ but just one equation, which will be solved using turbulence models.

However, for $\mathrm{Re}_\tau \to \infty$, $\tau_t^+ = 1$. Thus, all momentum is transferred by the turbulence, being the viscous momentum transfer negligible. However, this does not satisfy the boundary condition. Hence, this implies that for high Reynolds flows, there will be a two-layer structure: the core layer with $\tau_t^+ \sim 1$, in which the viscous transfer will be negligible; and the thin wall layer (or sublayer), in which both transfer methods are present.

The thickness of the thin wall layer, δ_ν, is determined by the viscosity ν and the wall friction velocity using (4.41).

$$\delta_\nu = \frac{\nu}{u_\tau} = \frac{H}{\mathrm{Re}_\tau} \tag{4.41}$$

For high Reynolds, hence, it tends to zero, which implies that the wall processes do not depend on H, being this the reason why this development can be applied to a general case. Thus, it is natural to define a stretched wall coordinate y^+, defined by (4.42).

$$y^+ = \frac{y}{\delta_\nu} = \frac{y u_\tau}{\nu} = \eta \mathrm{Re}_\tau \tag{4.42}$$

Introducing (4.42) in (4.40) yields

$$\frac{du^+}{dy^+} + \tau_t^+ = 1 \tag{4.43}$$

where, at $y^+ = 0, du^+/dy^+ = 1, u^+ = 0$. Thus, close to the wall, in the called pure viscous sublayer, the wall units will follow (4.44).

$$u^+ = y^+ \tag{4.44}$$

Nevertheless, in all the range for the thin wall layer $u^+ = f(y^+)$.

Logarithmic overlap law

Thus, the behaviour at the pure viscous sublayer and the core layer is known. Nonetheless, these layers have to match somehow, which is called to be the overlap layer, on s'ha de complir (4.45).

$$\hat{y}\frac{du^+}{d\hat{y}} = \frac{1}{\kappa} \tag{4.45}$$

where $\hat{y} = \eta \mathrm{Re}_\tau^\alpha$ behaves as an intermediate coordinate in the overlap layer, where, at $\alpha = 1, \hat{y} = y^+$ and $\alpha = 0, \hat{y} = \eta$; and κ is the von Kármán constant, which value is $\kappa = 0.41$. Thus, at the limit to $\eta \to 0$ and $y^+ \to \infty$ it has to match. For the wall layer, thus,

$$\lim_{y^+ \to \infty} \frac{du^+}{dy^+} = \frac{1}{\kappa y^+}$$

which, integrated, yields (4.46), which integration constant $C^+ = 5.0$ for smooth walls.

$$\lim_{y^+ \to \infty} u^+(y^+) = \frac{1}{\kappa} \ln y^+ + C^+ \tag{4.46}$$

Universal Law of the Wall

Using an indirect turbulence model, Gersten and Herwig [14] found an approximation to the unversal law of the wall in (4.47)-(4.48).

$$\frac{du^+}{dy^+} = \frac{1}{1 + (A+B)(y^+)^3} + \frac{B(y^+)^3}{1 + \kappa B (y^+)^4} \tag{4.47}$$

$$u^+ = \frac{1}{\Lambda}\left[\frac{1}{3}\ln\frac{\Lambda y^+ + 1}{\sqrt{(\Lambda y^+)^2 - \Lambda y^+ + 1}} + \frac{1}{\sqrt{3}}\left(\arctan\frac{2\Lambda y^+ - 1}{\sqrt{3}} + \frac{\pi}{6}\right)\right] + \frac{1}{4\kappa}\ln 1 + \kappa B(y^+)^4 \tag{4.48}$$

where $\kappa = 0.41$,$A = 6.1 \cdot 10^{-4}$,$B = 1.43 \cdot 10^{-3}$,$\Lambda = (A + B)^{1/3}$,$C^+ = 5.0$, yielding Figure 4.6.

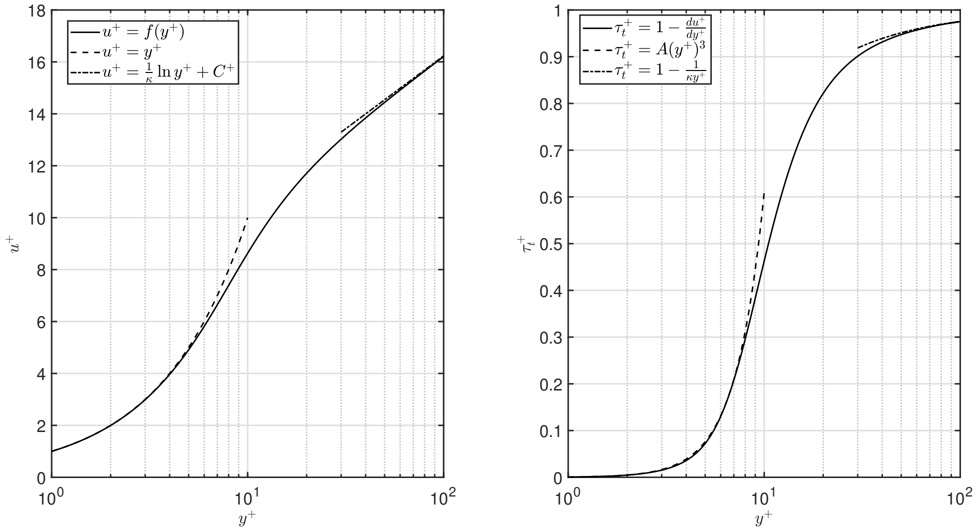


Figure 4.6: Universal law for u^+ and τ_t^+ considering a smooth wall ($C^+ = 5.0$). Adapted from [13].

Thus, the wall layer is now fully defined as Table 4.1.

Sublayer	y^+ range	$u^+ = f(y^+)$
Pure viscous sublayer	$0 \leq y^+ < 5$	$u^+ = y^+$
Buffer layer	$5 < y^+ < 70$	Eq.(4.48)
Overlap layer	$70 < y^+$	$u^+ = 1/\kappa \ln y^+ + C^+$

Table 4.1: Regions in the wall viscous sublayer. Adapted from [13].

Chapter 5

Turbulence. Large Eddy Simulation

In this chapter, the terms u_i, p_i will stand for the dimensionless magnitudes instead of the dimension magnitudes.

The Large Eddy Simulation relies on a low-pass frequency filtering (Figure 5.1) for the scales simulated. Thus, from the energy cascade, the large and mid scales are fully simulated while the smallest scales are the aim of the models applied. Thus, the model will add an extra energy dissipation required to fulfill the conservation laws without requiring these scales - hence allowing the control volumes to be bigger than η.

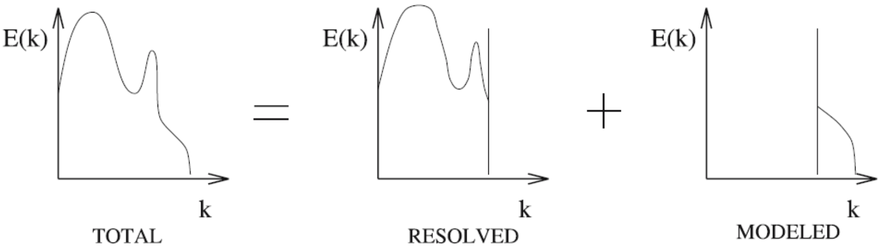


Figure 5.1: Decomposition of the energy spectrum related to the LES. Extracted from [15].

5.1 Introduction to function filtering

Thus, in order to separate the scales a filter has to be applied. In fact, this filter can be expressed as a convolution product (5.1),

$$\tilde{\phi}(x_i,t) = \int_{\mathbb{R}^4} \left(\phi(\xi_i,t')\mathrm{G}(x_i-\xi_i,t-t')\ dt'\ d^3\xi\right) \tag{5.1}$$

in which G stands for the convolution filter used, which will be indeed associated to the cutoff scales in space $\tilde{\Delta}$ and time τ_c. In a reduced notation, (5.1) will be noted as (5.2).

$$\tilde{\phi} = \mathrm{G} \star \phi \tag{5.2}$$

Converted onto Fourier space, the relationship yields

$$\hat{\tilde{\phi}} = \hat{\phi}\hat{\mathrm{G}}$$

where $\hat{G}$ corresponds to the transfer function of the convolution kernel. Thus, in this space there will be their cutoff wavenumber k_c associated to the cutoff length scale, as well as a cutoff frequency ω_c, associated to the cutoff time scale.

The unresolved part (thus, the fraction filtered) may be computed as

$$\phi'(x_i,t) = \phi(x_i,t) - \tilde{\phi}(x_i,t) = \phi(x_i,t) - \mathrm{G} \star \phi(x_i,t)$$

$$\phi'(x_i,t) = (1-\mathrm{G}) \star \phi(x_i,t) \tag{5.3}$$

which, in Fourier subspace yields $\hat{\phi}' = (1-\hat{\mathrm{G}})\hat{\phi}$

This filter, however, will have to verify three different properties to be appliable to Navier-Stokes equations [15]. First of all, will have to conserve the constants (5.4)

$$\tilde{a} = a \Longleftrightarrow \int_{\mathbb{R}^4} \mathrm{G}\ dt\ d^3\xi = 1 \tag{5.4}$$

Thus, the filter will have to be space-and-time bounded. and scaled to have an integral equal to 1. Moreover, it has to follow the linearity property $\widetilde{\phi+\psi} = \tilde{\phi} + \tilde{\psi}$, which can be demonstrated easily by the summation of integrals. Eventually, it has be commutative with derivation. Thus,

$$\widetilde{\frac{\partial \phi}{\partial s}} = \frac{\partial \tilde{\phi}}{\partial s},\ s = x_i, t,\ i = 1:3$$

At this point, the commutator operator may be defined as $[f,g]$ (5.5) where f and g are two operators applied on ϕ.

$$[f,g]\phi = (f \circ g)\phi - (g \circ f)\phi = f(g(\phi)) - g(f(\phi)) \tag{5.5}$$

The commutator operator properties, which are trivial to proof using the definition, are the following:

1. Skew-symmetry $[f,g] = -[g,f]$
2. Leibniz identity $[f \circ g, h] = [f,h] \circ g + f \circ [g,h]$
3. Jacobi's identity $[f,[g,h]] + [g,[h,f]] + [h,[f,g]] = 0$

Thus, applied to the case in concern, it follows that

$$\left[G\star, \frac{\partial}{\partial s}\right] = 0$$

Moreover, in its general form the filter G is not a projector, since $\mathrm{G} \circ \mathrm{G} \neq \mathrm{G}$, which are defined as $P^n = P, \forall n \in \mathbb{N}^+$. Thus, in this general case the filter can be interpreted as a change of variables and, hence, it can be inverted without any loss of information. However, in the particular case in which $G = 1$, it will, in fact, be a projector and, therefore, its inverse will imply information loss.

Now, it the convolution kernel is assumed to be factorized from a four-dimensional to a product of mono-dimensional kernels, (5.6) follows.

$$\mathrm{G}(x_i - \xi_i, t - t') = G_t(t-t') \prod_{j=1}^{3} G_j(x_j - \xi_j) \tag{5.6}$$

Since the most common technique in LES is space-filtering, the temporal filter can be expressed as

$$G_t(t-t') = \delta(t-t')$$

where δ is the Dirac function. However, the space kernels may be very varied.

It is important to recall, nonetheless, that the space-filtering implies an implicit time-filter:

Let $k_c = \frac{\pi}{\tilde{\Delta}}$ and $E(k)$ be the energy spectrum of the exact solution. Then, the kinetic energy may be expressed as $E_k = kE(k)$. At the cutoff wavenumber, $E_{k_c} = k_c E(k_c)$. Thus, the velocity can be scaled as $v_c = \sqrt{k_c E(k_c)}$.

Now, defining the timescale as

$$\tau_c = \frac{\tilde{\Delta}}{v_c} = \frac{\tilde{\Delta}}{\sqrt{k_c E(k_c)}}$$

where k_c depends on $\tilde{\Delta}$ as previously defined.

Classical filters

Among all the sets of filters available, some categories arise: symmetric, homogeneous, etc. However, three classical homogeneous and symmetric filters are presented, even though the third one will not be valid since it will be defined as a projector.

1. Box filter. Characterized by (5.7) as its convolution kernels in the original domain and Fourier subspace. Its representation corresponds to Figure 5.2.

$$\mathrm{G}(x-\xi) = \begin{cases} \frac{1}{\tilde{\Delta}} & \text{if } |x-\xi| \le \frac{\tilde{\Delta}}{2} \\ 0 & \text{otherwise} \end{cases} \tag{5.7a}$$

$$\hat{\mathrm{G}} = \frac{\sin k\tilde{\Delta}/2}{k\tilde{\Delta}/2} \tag{5.7b}$$

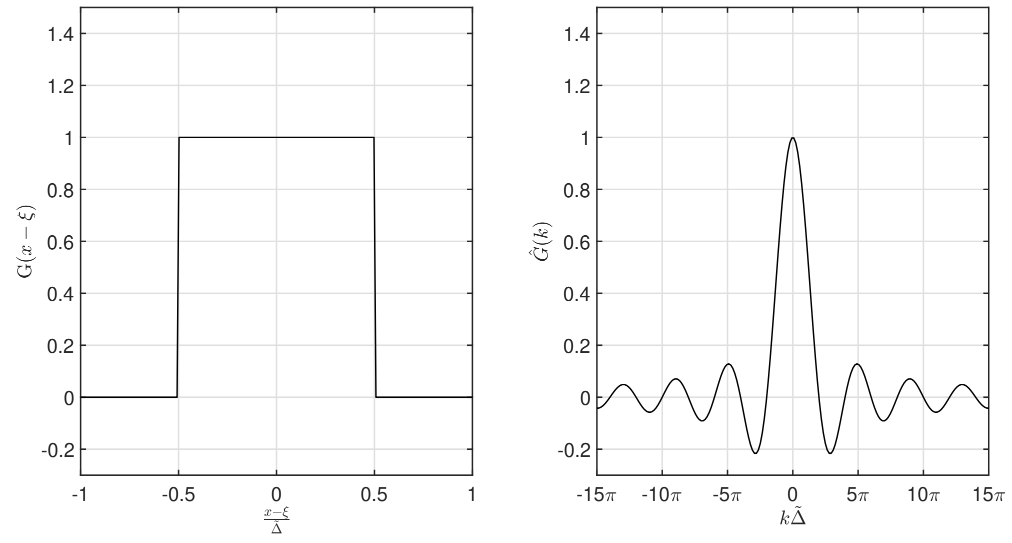


Figure 5.2: Representation of the box filter kernel in the original domain (left) and in Fourier subspace (right).

2. Gaussian filter. Characterized by (5.8) as its convolution kernels, where usually $\gamma = 6$. in the original domain and Fourier subspace. Its graphical representation corresponds to Figure 5.3.

$$\mathrm{G}(x-\xi) = \left(\frac{\gamma}{\pi\tilde{\Delta}^2}\right)^{1/2} \exp\left(\frac{-\gamma|x-\xi|^2}{\tilde{\Delta}^2}\right) \tag{5.8a}$$

$$\hat{\mathrm{G}} = \exp\left(\frac{-\tilde{\Delta}^2 k^2}{4\gamma}\right) \tag{5.8b}$$

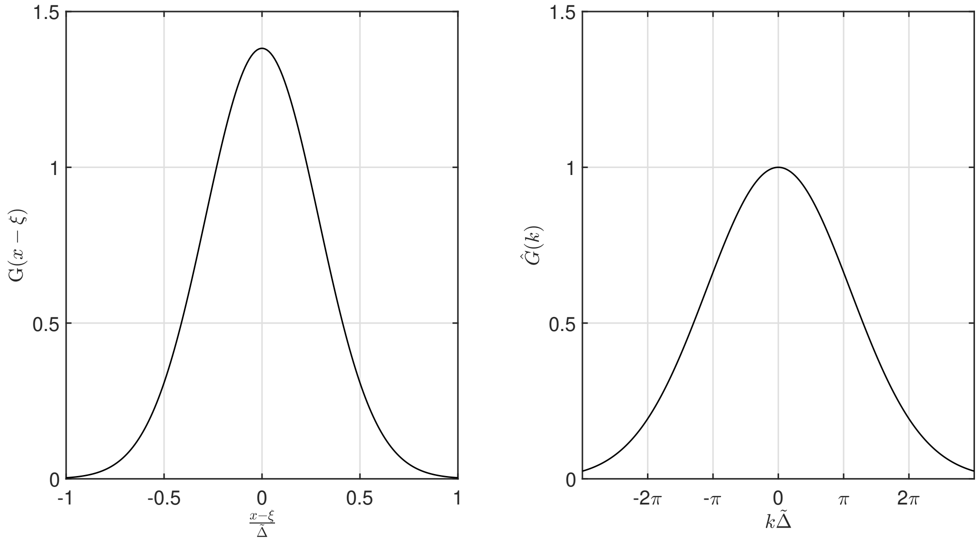


Figure 5.3: Representation of the gaussian filter kernel in the original domain (left) and in Fourier subspace (right).

5.2 Application of the method to Navier-Stokes equations

Applying the filter in its general form to the Navier-Stokes equations, considering the properties previously detailed, (5.9) is found.

$$\frac{\partial \tilde{u}_i}{\partial x_i} = 0 \tag{5.9a}$$

$$\frac{\partial \tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_j}(\widetilde{u_i u_j}) = -\frac{\partial \tilde{p}}{\partial x_i} + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x_j}\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right) \tag{5.9b}$$

The main concern is the nonlinear term in the momentum equation. Thus, it will be applied that $u_i = \tilde{u}_i + u'_i$. Expanding the product terms, (5.10) follows, which is known as the double decomposition.

$$\widetilde{u_i u_j} = \widetilde{\tilde{u}_i \tilde{u}_j} + \underbrace{\underbrace{\widetilde{\tilde{u}_i u'_j + u'_i \tilde{u}_j}}_{C_{ij}} + \underbrace{\widetilde{u'_i u'_j}}_{R_{ij}}}_{T_{ij}} \tag{5.10}$$

Where the subgrid tensor T_{ij} firstly appears. The name double decomposition relies on this subgrid tensor having two main components, the cross-stress tensor C_{ij}, which concerns the interaction between the resolved and the subgrid scales; and the Reynolds subgrid tensor R_{ij}, in regards to the interactions between the subgrid scales. Thus, applied onto (5.9) it can be found the solved momentum equation for the double decomposition.

$$\frac{\partial \tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_j}(\widetilde{\tilde{u}_i\tilde{u}_j}) = -\frac{\partial \tilde{p}}{\partial x_i} + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x_j}\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right) - \frac{\partial T_{ij}}{\partial x_j} \tag{5.11}$$

Nonetheless, Leonard stated that the nonlinear term should be calculated using just the filtered variables, without the double filter in the double decomposition case. Thus, $\widetilde{\tilde{u}_i\tilde{u}_j} = \widetilde{\tilde{u}_i\tilde{u}_j} + \tilde{u}_i\tilde{u}_j - \tilde{u}_i\tilde{u}_j$. This implies the appearance of a third tensor, the Leonard tensor L_{ij} in (5.12).

$$\widetilde{u_i u_j} = \tilde{u}_i\tilde{u}_j + \underbrace{\underbrace{\widetilde{\tilde{u}_i\tilde{u}_j} - \tilde{u}_i\tilde{u}_j}_{L_{ij}} + \underbrace{\widetilde{\tilde{u}_i u'_j + u'_i\tilde{u}_j}}_{C_{ij}} + \underbrace{\widetilde{u'_i u'_j}}_{R_{ij}}}_{T_{ij}} \tag{5.12}$$

Thus, the momentum equation related to this decomposition corresponds to (5.13).

$$\frac{\partial \tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_j}(\tilde{u}_i\tilde{u}_j) = -\frac{\partial \tilde{p}}{\partial x_i} + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x_j}\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right) - \frac{\partial T_{ij}}{\partial x_j} \tag{5.13}$$

This approach, known as the Leonard (or triple) decomposition, allows the nonlinear term to be computed with the same fineness as the linear terms [16]. This is the reason why the remainder of the development will be done under this approach.

Thus, the subgrid scale will be found by obtaining the difference between the full Navier-Stokes and (5.13) as well as the filtered continuity equation. Thus, it follows that, after operating algebraically [15], it corresponds to (5.14):

$$\frac{\partial u'_i}{\partial x_i} = 0 \tag{5.14a}$$

$$\frac{\partial u'_i}{\partial t} + \frac{\partial}{\partial x_j}((\tilde{u}_i + u'_i)(\tilde{u}_j + u'_j) - \tilde{u}_i\tilde{u}_j) = -\frac{\partial p'}{\partial x_i} + \frac{\partial T_{ij}}{\partial x_j} + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x_j}\left(\frac{\partial u'_i}{\partial x_j} + \frac{\partial u'_j}{\partial x_i}\right) \tag{5.14b}$$

5.3 Eddy-viscosity assumption

The eddy-viscosity model considers that the subgrid stress tensor can be modelled as a function of a model parameter, the eddy viscosity ν_t, as well as the filtered rate of strain $\tilde{S}_{ij} = 1/2(\partial_j\tilde{u}_i + \partial_i\tilde{u}_j)$. Thus, the subgrid stress tensor will be defined as (5.15).

$$T_{ij} = 2\nu_t\tilde{S}_{ij} \tag{5.15}$$

The model now will be based on how the eddy viscosity is modelled, as previously explained. Thus, the LES equations will be (5.16).

$$\frac{\partial \tilde{u}_i}{\partial x_i} = 0 \tag{5.16a}$$

$$\frac{\partial \tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_j}(\tilde{u}_i\tilde{u}_j) = -\frac{\partial \tilde{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\left(\frac{1}{\mathrm{Re}} + \nu_t\right)\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right)\right] \tag{5.16b}$$

5.4 Smagorinsky model

The most used eddy-viscosity model is, in fact, Smagorinsky model [17], which states that the eddy-viscosity is proportional to a characteristic subgrid scale velocity [16], defined as (5.17)

$$v_\Delta = \tilde{\Delta}\sqrt{2\tilde{S}_{ij}\tilde{S}_{ij}} \tag{5.17}$$

In fact, the Smagorinsky's eddy viscosity [16] corresponds to (5.18).

$$\nu_t = (C_S\tilde{\Delta})^2\sqrt{2\tilde{S}_{ij}\tilde{S}_{ij}} \tag{5.18}$$

where C_S is the Smagorinsky constant, which can be calculated, approximately, as $C_S \approx \frac{1}{\pi}\left(\frac{3C_K}{2}\right)^{-3/4}$. This, for $C_K = 1.4$ yields $C_S \approx 0.18$, which is generally acceptable for isotropic turbulence. In the case of shear flows, nevertheless, it is more preferable to use $C_S = 0.1$ [16]. Thus, C_S is not, indeed, a universal constant.

The rate of energy transfer from the resolved scales to the subgrid scales [2] corresponds to

$$\mathcal{P} = -T_{ij}\tilde{S}_{ij} = 2\nu_t\tilde{S}_{ij}\tilde{S}_{ij} \tag{5.19}$$

Thus, in $\nu_t > 0$, the energy cascade will be as expected by Kolmogorov's theory and there will not be a backscatter of energy [2].

5.5 Burgers' equation

In chapter 4 a DNS was done on the Burgers' equation (4.28). In order to obtain rather good results for Re = 40, a cutoff wavenumber of $k_N = 100$ was required.

In this section, however, the spectral LES model (5.20) will be applied with $C_K = 0.4523$, which corresponds to the optimal value, and $C_K = 0.05$, which corresponds to a very poor value; as well as $m = 2$.ç

$$\nu_t(k/k_N) = \nu_t^{+\infty}\left(\frac{E_{kN}}{k_N}\right)^{1/2}\nu_t^*\left(\frac{k}{k_N}\right) \tag{5.20}$$

where

$$\nu_t^{+\infty} = 0.31\frac{5-m}{m+1}\sqrt{3-m}C_K^{-3/2} \tag{5.21a}$$

$$\nu_t^*\left(\frac{k}{k_N}\right) = 1 + 34.5\exp\left(-3.03(k_N/k)\right) \tag{5.21b}$$

Thus, a parameter called ν_{eff} will be introduced at each k, considering that

$$\nu_{eff} = \frac{1}{\text{Re}} + \nu_t$$

.

First of all, the comparison between the LES resolution for a cutoff wavenumber of 20, with the Kolmogorov constants previously detailed, with the DNS solution at $k_N = 100$ and $k = 20$ is displayed in Figure 5.4, which shows that the introduction of a proper LES model implies that a solution for the problem can be obtained with a much coarser domain.

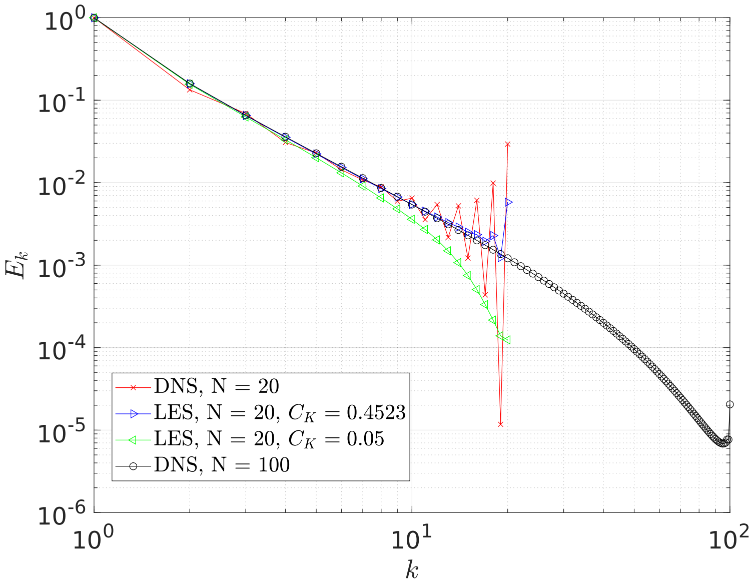


Figure 5.4: Results for different studied cases of the resolution of the Burgers' Equation.

In fact, not just a proper C_K is required. In this case, since m is also a parameter in the model, its implication has to be studied. Thus, Figure 5.5 shows that a proper m will have the model to be right on the DNS or differ a lot from the actual solution.

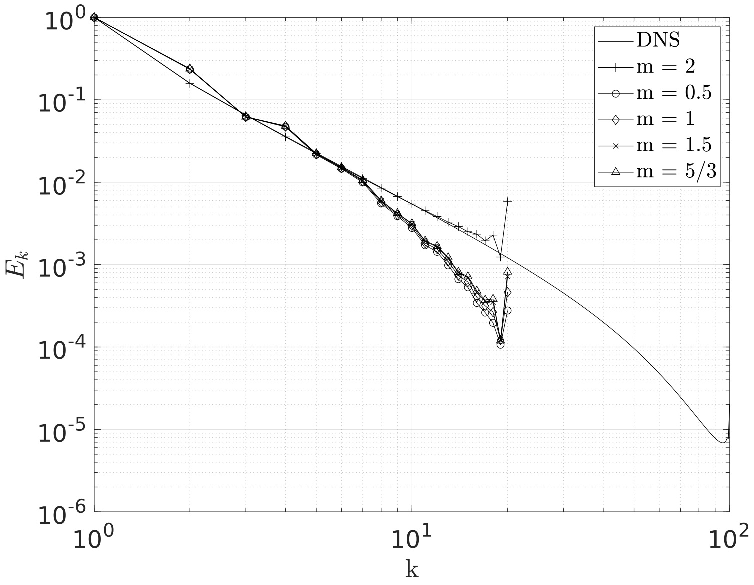


Figure 5.5: Influence of the parameter m on the LES solution for Re = 40.

5.6 Invariant framework of LES

The models mostly rely on the use of five different invariants [18] (six, in case of a compressible flow) all related to the second-order gradient tensor of the velocity field $G_{ij} = \frac{\partial u_i}{\partial x_j}$, and the symmetric (rate of strain, S_{ij}) and skew-symmetric $\Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right)$ parts of the tensor.

These invariants (5.22) are the following: Q_G, R_G, Q_S, R_S, V^2, where, Q_A and R_A are the second and third invariants (being the trace P_A the first), respectively, of a second-order tensor A.

$$Q_A = \frac{1}{2}[\mathrm{tr}^2(A) - \mathrm{tr}\,(A^2)] \tag{5.22a}$$

$$R_A = \frac{1}{6}[\mathrm{tr}^3(A) - 3\,\mathrm{tr}\,(A)\,\mathrm{tr}\,(A^2) + 2\,\mathrm{tr}\,(A^3)] \tag{5.22b}$$

$$V^2 = 4(\mathrm{tr}\,(S^2\Omega^2) - 2Q_S Q_\Omega) \tag{5.22c}$$

Considering that all tensors G, S, Ω are traceless, (5.22) is reduced as follows:

$$Q_A = -\frac{1}{2}\,\mathrm{tr}\,(A^2) \tag{5.23a}$$

$$R_A = \frac{1}{3}\,\mathrm{tr}\,(A^3) \tag{5.23b}$$

These invariants determine, in its own measure, the performance of the model in all regions, being the most critical the near wall region. Thus, its use will determine its wall-behaviour Table 5.1.

Invariant	Q_G	R_G	Q_S	R_s	V^2
Wall-behaviour	$\mathcal{O}(y^2)$	$\mathcal{O}(y^3)$	$\mathcal{O}(y^0)$	$\mathcal{O}(y^1)$	$\mathcal{O}(y^2)$

Table 5.1: Near-wall behaviour of the five basic invariants. Adapted from [18].

5.6.1 Usual LES models under the invariant framework

Smagorinsky model

As previously defined, Smagorinsky's eddy viscosity model can be defined as $(C_S\Delta)^2|S|$. Nonetheless, under this framework, it can be redefined as (5.24), considering that $|S| = (2S_{ij}S_{ij})^{1/2} = (2\,\mathrm{tr}\,(S^2))^{1/2} = 2(-Q_S)^{1/2}$.

$$\nu_t^S = 2(C_S\Delta)^2(-Q_S)^{1/2} \tag{5.24}$$

Thus, as it can be seen in Table 5.1, the use of Q_S as the only invariant in the method implies that its near-wall behaviour corresponds to $\mathcal{O}(y^0)$, being this the reason why it cannot be used when modelling near-wall flows.

WALE model

Firstly defined by Nicoud and Ducros (see [19] for further details in the development), it is based on the second invariant of the traceless part of the symmetric tensor, yielding the WALE (Wall-adapting local eddy) model (5.25).

$$\nu_t^W = (C_W\Delta)^2 \frac{(V^2/2 + 2Q_G^2/3)^{3/2}}{(-2Q_S)^{5/2} + (V^2/2 + 2Q_G^2/3)^{5/4}} \tag{5.25}$$

Due to the fact that it uses a great number of invariants, its implementation is one of the most expensive among the common LES models. Nonetheless, its use is justified due to an excellent near-wall behaviour ($\mathcal{O}(y^3)$).

Vreman's model and S3PQR models

Based on the ration between the second and first invariants of the tensor GG^*. Following the development in [18], it is derived that

$$\frac{Q_{GG^*}}{P_{GG^*}} = \frac{V^2 + Q_G^2}{2(Q_\Omega - Q_S)}$$

Thus, Vreman's model yields (5.26)

$$\nu_t^{Vr} = (C_{Vr}\Delta)^2 \left(\frac{V^2 + Q_G^2}{2(Q_\Omega - Q_S)} \right)^{1/2} \tag{5.26}$$

Due to its simplicity, which yields a near-wall behaviour of $\mathcal{O}(y)$, Vreman's model is a well-conditioned model.

Trias et al. [18] used a similar approach in considering models based on the invariants of GG^*, yielding what is known as S3PQR models, which take (5.27) as its general form.

$$\nu_t^{S3PQR} = (C_{S3PQR}\Delta)^2 P_{GG^*}^p Q_{GG^*}^q R_{GG^*}^r \tag{5.27}$$

It is derived in [18] that, for (5.27), in order to have frequency units, the general form must be reduced to

$$\nu_t^{S3PQR} = (C_{S3PQR}\Delta)^2 P_{GG^*}^p Q_{GG^*}^{-(p+1)} R_{GG^*}^{(p+5/2)/3}$$

In this case, the performance of the model in regards to near-wall behaviour is better ($\mathcal{O}(y^3)$) than Vreman's model, while maintaining a quite simple implementation as well as a rather low computational cost.

Verstappen's model

Verstappen proposed a third-invariant based eddy viscosity model (5.28), just depending on the symmetric part, which has a similar behaviour in terms of wall beahviour as well as in computational cost.

$$\nu_t^{Ve} = (C_{Ve}\Delta)^2 \frac{|R_S|}{-Q_S} \tag{5.28}$$

Chapter 6

Numerical solution of turbulence

6.1 Symmetry-preserving discretization

As follows from the turbulence theory, the turbulent phenomena consists in the combination of nonlinear transport and dissipation of energy.

Nonetheless, this energy is just dissipated by the viscous forces. Thus, in an inviscid flow, the total amount of energy should be maintained. Nonetheless, if there is dissipation, these levels of energy will not just be maintained but instead get lower.

Thus, if the time and space are discretized keeping the symmetries of the differential operators, the method will be unconditionally stable and conservative [20].

It is important to remind that turbulence is essentially a tridimensional phenomenon. Thus, from here on the simulations will be done on tridimensional conditions.

6.1.1 Operator symmetries

The operators appearing in the Navier-Stokes equations are the divergence $\nabla\cdot$, the gradient ∇, the laplacian ∇^2 and the convective operator $C(\vec{u},\phi) = (\vec{u}\cdot\nabla)\phi$.

Introducing the inner product of functions as

$$\langle a|b\rangle = \int_\Omega ab\ dV$$

, some properties of the operators can be shown [21] by assuming no contribution in any case from the domain boundary.

$$\langle \nabla\cdot\vec{a}|\phi\rangle = -\langle \vec{a}|\nabla\phi\rangle \tag{6.1a}$$

$$\langle \nabla^2 f|g\rangle = -\langle \nabla f|\nabla g\rangle = \langle f|\nabla^2 g\rangle \tag{6.1b}$$

$$\langle C(\vec{u},\phi_1)|\phi_2\rangle = -\langle C(\vec{u},\phi_2)|\phi_1\rangle \tag{6.1c}$$

Thus, the discretization of each of these operators will have to keep these properties so as to simulate the turbulent phenomena as good as possible provided that all symmetries will be maintained.

First of all, it is important to define the reference frame from which the discretization is going to be developed, which is outlined in Figure 6.1

Nonetheless, why is it important to preserve the operators' properties?

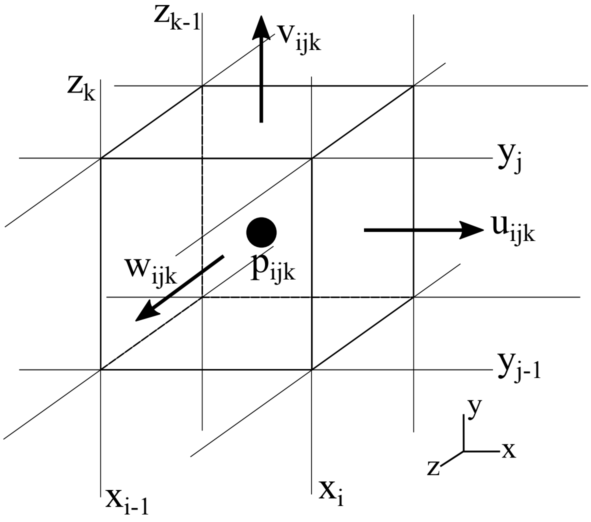


Figure 6.1: Reference frame of the discretization. Main control volume

Let us consider a discretized convection-diffusion equation [20], without any source term (6.2), in which $\bar{\mathbf{u}}$ is assumed to be constant.

$$\mathbf{\Omega}\frac{d\mathbf{u}_h}{dt} + \mathbf{C}(\bar{\mathbf{u}})\mathbf{u}_h + \mathbf{D}\mathbf{u}_h = \mathbf{0} \tag{6.2}$$

Thus, assuming $\mathbf{D} = \mathbf{0}$ (inviscid flow), the transport equation for kinetic energy becomes (6.3). Thus, the lack of viscosity should imply that the kinetic energy remains constant, which will only happen, if and only if, $\mathbf{C}$ is skew-symmetric.

$$\frac{d}{dt}||\mathbf{u}_h||^2 = -\mathbf{u}_h^*(\mathbf{C} + \mathbf{C}^*)\mathbf{u}_h \tag{6.3}$$

Thus, by using a skew-symmetric convective operator there will not be any additional energy damping other than the viscosity itself, which is, in fact, inherited from the differential operator (6.1c).

6.1.2 Convective operator discretization

Applying the Reynolds' transport theorem (6.4) applied on u discretized on a x-staggered control volume, the first approach to the convective operator can be found in (6.5), being Figure 6.2 a bidimensional outline of the control volume. Let $\bar{u}_{i,j,k}$ the mass flow through the control volume boundaries.

$$\frac{d}{dt}\int_{\Omega} f \ dV = \int_{\Omega} \frac{\partial f}{\partial t} \ dV + \int_{\partial\Omega} f\vec{u} \cdot \hat{n} \ dA \tag{6.4}$$

Then,

$$|\Omega_{i+1/2,j,k}|\frac{du_{i,j,k}}{dt} + \bar{u}_{i+1/2,j,k}u_{i+1/2,j,k} + \bar{v}_{i+1/2,j,k}u_{i,j+1/2,k} + \bar{w}_{i+1/2,j,k}u_{i,j,k+1/2} -$$

$$- \bar{u}_{i-1/2,j,k}u_{i-1/2,j,k} - \bar{v}_{i+1/2,j-1,k}u_{i,j-1/2,k} - \bar{w}_{i+1/2,j,k-1}u_{i,j,k-1/2} \tag{6.5}$$

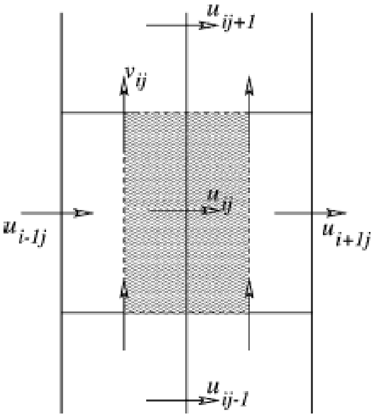


Figure 6.2: Bidimensional representation of the control volume $\Omega_{i+1/2,j,k}$. Extracted from [20].

At the staggered control volume faces, the velocity will be approximated using a Central Difference Scheme:

$$u_{i+1/2,j,k} = \frac{1}{2}(u_{i+1,j,k} + u_{i,j,k})$$

Thus, adding these definitions onto (6.5) and operating:

$$\begin{aligned}|\Omega_{i+1/2,j,k}|\frac{du_{i,j,k}}{dt}+\frac{1}{2}(\bar{u}_{i+1/2,j,k}+\bar{v}_{i+1/2,j,k}+\bar{w}_{i+1/2,j,k}-\bar{u}_{i-1/2,j,k}-\bar{v}_{i+1/2,j-1,k}-\bar{w}_{i+1/2,j,k-1})u_{i,j,k}+\\ +\frac{1}{2}\bar{u}_{i+1/2,j,k}u_{i+1,j,k}+\frac{1}{2}\bar{v}_{i+1/2,j,k}u_{i,j+1,k}+\frac{1}{2}\bar{w}_{i+1/2,j,k}u_{i,j,k+1}-\\ -\frac{1}{2}\bar{u}_{i-1/2,j,k}u_{i-1,j,k}-\frac{1}{2}\bar{v}_{i+1/2,j-1,k}u_{i,j-1,k}+\frac{1}{2}\bar{w}_{i+1/2,j,k-1}u_{i,j,k-1}\end{aligned} \tag{6.6}$$

Applying the same methodology to the other two directions, y and z, (6.7) and (6.8), respectively, are obtained.

$$\begin{aligned}|\Omega_{i,j+1/2,k}|\frac{dv_{i,j,k}}{dt}+\frac{1}{2}(\bar{u}_{i,j+1/2,k}+\bar{v}_{i,j+1/2,k}+\bar{w}_{i,j+1/2,k}-\bar{u}_{i-1,j+1/2,k}-\bar{v}_{i,j-1/2,k}-\bar{w}_{i,j+1/2,k-1})v_{i,j,k}+\\ +\frac{1}{2}\bar{u}_{i,j+1/2,k}v_{i+1,j,k}+\frac{1}{2}\bar{v}_{i,j+1/2,k}v_{i,j+1,k}+\frac{1}{2}\bar{w}_{i,j+1/2,k}v_{i,j,k+1}-\\ -\frac{1}{2}\bar{u}_{i-1,j+1/2,k}v_{i-1,j,k}-\frac{1}{2}\bar{v}_{i,j-1/2,k}v_{i,j-1,k}-\frac{1}{2}\bar{w}_{i,j+1/2,k-1}v_{i,j,k-1}\end{aligned} \tag{6.7}$$

$$\begin{aligned}|\Omega_{i,j,k+1/2}|\frac{dw_{i,j,k}}{dt}+\frac{1}{2}(\bar{u}_{i,j,k+1/2}+\bar{v}_{i,j,k+1/2}+\bar{w}_{i,j,k+1/2}-\bar{u}_{i-1,j,k+1/2}-\bar{v}_{i,j-1,k+1/2}-\bar{w}_{i,j,k-1/2})w_{i,j,k}+\\ +\frac{1}{2}\bar{u}_{i,j,k+1/2}w_{i+1,j,k}+\frac{1}{2}\bar{v}_{i,j,k+1/2}w_{i,j+1,k}+\frac{1}{2}\bar{w}_{i,j,k+1/2}w_{i,j,k+1}-\\ -\frac{1}{2}\bar{u}_{i-1,j,k+1/2}w_{i-1,j,k}-\frac{1}{2}\bar{v}_{i,j-1,k+1/2}w_{i,j-1,k}-\frac{1}{2}\bar{w}_{i,j,k-1/2}w_{i,j,k-1}\end{aligned} \tag{6.8}$$

In a general form, (6.6),(6.7) and (6.8) can be synthesized in (6.9)

$$\mathbf{\Omega}_1 \frac{d\mathbf{u}_h}{dt} + \mathbf{C}_1(\bar{\mathbf{u}})\mathbf{u}_h \tag{6.9}$$

where

$$\mathbf{u}_h = (u_{1,1,1}, \dots, u_{i,j,k}, \dots, u_{n_x,n_y,n_z}, v_{1,1,1}, \dots, v_{i,j,k}, \dots, v_{n_x,n_y,n_z}, \dots, w_{1,1,1}, \dots, w_{i,j,k}, \dots, w_{n_x,n_y,n_z})^*$$

$$\mathbf{\Omega}_1 = \mathrm{diag}(|\Omega_{i+1/2,j,k}|, |\Omega_{i,j+1/2,k}|, |\Omega_{i,j,k+1/2}|) \;\; \forall i,j,k$$

and $\mathbf{C}_1$ is the convective operator which has to fulfill $\mathbf{C}_1 + \mathbf{C}_1^* = \mathbf{0}$, given its skew-symmetry [20, 22]. In order to do so, the diagonal terms have to be equal to zero, which is accomplished since the velocity approximation is done using a CDS as well as by mass conservation. This implies that the interface mass fluxes has to be calculated using (6.10) (and its equivalents for all directions) [20].

$$\bar{u}_{i+1/2,j,k} = \frac{1}{2}(\bar{u}_{i,j,k} + \bar{u}_{i+1,j,k}) \tag{6.10}$$

6.1.3 Divergence and gradient operators discretization

Obviously, the mass flux has to be computed as a function of the discrete velocity $\mathbf{u}_h$, being then $\mathbf{C}_1(\bar{\mathbf{u}}) = \mathbf{C}_1(\bar{\mathbf{u}}(\mathbf{u}_h))$. Thus, the integration of the mass flux in each direction using the mid-point rule. Eq. (6.11) states the methodology in u, nonetheless, its extension in v and w is straightforward.

$$\bar{u}_{i,j,k} = (y_j - y_{j-1})(z_k - z_{k-1})u_{i,j,k} \tag{6.11}$$

Thus, in each main grid cell, mass conservation $\nabla \cdot \vec{u} = 0$ gives (6.12)

$$\bar{u}_{i,j,k} + \bar{v}_{i,j,k} + \bar{w}_{i,j,k} - \bar{u}_{i-1,j,k} - \bar{v}_{i,j-1,k} - \bar{w}_{i,j,k-1} = 00 \tag{6.12}$$

Thus, applying (6.11) to (6.12), and using the compact form, mass conservation yields (6.13), being $\mathbf{M}_1$ the discrete form of the divergence operator.

$$\mathbf{M}_1\mathbf{u}_h = \mathbf{0} \tag{6.13}$$

In order to discretize the gradient operator, (6.14) will be used [20], which relates it with the divergence operator being $\mathbf{G}_1$ this gradient operator.

$$\mathbf{G}_1 = -\mathbf{\Omega}_1^{-1}\mathbf{M}_1^* \tag{6.14}$$

6.1.4 Diffusive operator discretization

The laplacian of an arbitrary field f can be expressed as

$$\frac{\partial}{\partial x}\frac{\partial f}{\partial x} + \frac{\partial}{\partial y}\frac{\partial f}{\partial y} + \frac{\partial}{\partial z}\frac{\partial f}{\partial z}$$

Let $\phi = \frac{1}{\mathrm{Re}}\partial_x u$, $\psi = \frac{1}{\mathrm{Re}}\partial_y u$ and $\nu = \frac{1}{\mathrm{Re}}\partial_z u$. Then, the integral on all the volume, applying Gauss' divergence theorem,

$$\int_\Omega \nabla \cdot \vec{f}\, dV = \int_{\partial\Omega} \vec{f} \cdot \hat{n}\, dA$$

In this case, let $\vec{f} = \nabla u$. Then, Gauss theorem enables the possibility of integrating at the interfaces of the control volumes. Thus, at each control volume it follows that (6.15) can be used for this purpose at each time t.

$$\begin{aligned}
&\int_{y_{j-1}}^{y_j}\int_{z_{k-1}}^{z_k} \phi(x_{i+1/2},y,z)\ dz\ dy - \int_{y_{j-1}}^{y_j}\int_{z_{k-1}}^{z_k} \phi(x_{i-1/2},y,z)\ dz\ dy+ \\
&+\int_{x_{i-1/2}}^{x_{i+1/2}}\int_{z_{k-1}}^{z_k} \psi(x,y_j,z)\ dz\ dx - \int_{x_{i-1/2}}^{x_{i+1/2}}\int_{z_{k-1}}^{z_k} \psi(x,y_{j-1},z)\ dz\ dx+ \\
&+\int_{x_{i-1/2}}^{x_{i+1/2}}\int_{y_{j-1}}^{y_j} \nu(x,y,z_k)\ dy\ dx - \int_{x_{i-1/2}}^{x_{i+1/2}}\int_{y_{j-1}}^{y_j} \nu(x,y,z_{k-1})\ dy\ dx
\end{aligned} \tag{6.15}$$

Following an equivalent notation to (6.11), each integral can be expressed as follows,

$$\int_{y_{j-1}}^{y_j}\int_{z_{k-1}}^{z_k} \phi(x_{i+1/2},y,z)\ dz\ dy = \bar{\phi}_{i+1/2,j,k} = (y_j - y_{j-1})(z_k - z_{k-1})\phi_{i+1/2,j,k}$$

where:

$$\begin{aligned}
\phi_{i+1/2,j,k} &= \frac{1}{\mathrm{Re}}\frac{u_{i+1,j,k}-u_{i,j,k}}{x_{i+1}-x_i} \\
\psi_{i,j,k} &= \frac{1}{\mathrm{Re}}\frac{u_{i,j+1,k}-u_{i,j,k}}{y_{j+1/2}-y_{j-1/2}} \\
\nu_{i,j,k} &= \frac{1}{\mathrm{Re}}\frac{u_{i,j,k+1}-u_{i,j,k}}{z_{k+1/2}-z_{k-1/2}}
\end{aligned}$$

This development yields the compact form

$$\bar{\phi}_{i+1/2,j,k} - \bar{\phi}_{i-1/2,j,k} + \bar{\psi}_{i,j,k} - \bar{\psi}_{i,j-1,k} + \bar{\nu}_{i,j,k} - \bar{\nu}_{i,j,k-1}$$

which, extended, provides the discretization of the laplacian for u component of the velocity field (6.17). The same exact procedure can be applied for both v and w components to find its respective fields.

$$\begin{aligned}
&(y_j - y_{j-1})(z_k - z_{k-1})\frac{1}{\mathrm{Re}}\frac{u_{i+1,j,k}-u_{i,j,k}}{x_{i+1}-x_i} - (y_j - y_{j-1})(z_k - z_{k-1})\frac{1}{\mathrm{Re}}\frac{u_{i,j,k}-u_{i-1,j,k}}{x_i-x_{i-1}}+ \\
&+(x_{i+1/2}-x_{i-1/2})(z_k-z_{k-1})\frac{1}{\mathrm{Re}}\frac{u_{i,j+1,k}-u_{i,j,k}}{y_{j+1/2}-y_{j-1/2}} - (x_{i+1/2}-x_{i-1/2})(z_k-z_{k-1})\frac{1}{\mathrm{Re}}\frac{u_{i,j,k}-u_{i,j-1,k}}{y_{j-1/2}-y_{j-3/2}}+ \\
&+(x_{i+1/2}-x_{i-1/2})(y_j-y_{j-1})\frac{1}{\mathrm{Re}}\frac{u_{i,j,k+1}-u_{i,j,k}}{z_{k+1/2}-z_{k-1/2}} - (x_{i+1/2}-x_{i-1/2})(y_j-y_{j-1})\frac{1}{\mathrm{Re}}\frac{u_{i,j,k}-u_{i,j,k-1}}{z_{k-1/2}-z_{k-3/2}}
\end{aligned} \tag{6.17}$$

Thus, all these developments can be synthesized on a compact matrix form $-\mathbf{D}_1\mathbf{u}_h$.

6.1.5 Fractional Step Method

After the proceedings of all previous sections, each and every operator of Navier-Stokes equations has been discretized, yielding the discrete Navier-Stokes equations (6.18).

$$\mathbf{M}_1\mathbf{u}_h = \mathbf{0} \tag{6.18a}$$

$$\boldsymbol{\Omega}_1\frac{d\mathbf{u}_h}{dt} + \mathbf{C}_1(\mathbf{u}_h)\mathbf{u}_h + \mathbf{D}_1\mathbf{u}_h - \mathbf{M}_1^*\mathbf{p}_h = \mathbf{0} \tag{6.18b}$$

Thus, the Fractional Step Method [23] as previously stated, uses a predicted velocity field to then project it onto the subspace with null divergence.

Let $\mathbf{u}_h^p$ be the predictor velocity, defined as

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

where $\mathbf{R}(\mathbf{u}_h) = -\boldsymbol{\Omega}_1^{-1}(\mathbf{C}_1(\mathbf{u}_h)\mathbf{u}_h + \mathbf{D}_1\mathbf{u}_h)$ and a Second Order Adams-Bashforth scheme is used to discretize the time dependance of $\mathbf{R}$. An adaptative method leading to a stable solution with longer timesteps can be seen in [24].

On the other hand, the current step velocity field can be defined as

$$\mathbf{u}_h^{n+1} = \mathbf{u}_h^p + \Delta_t\boldsymbol{\Omega}_1^{-1}\mathbf{M}_1^*\mathbf{p}_h^{n+1} \tag{6.19}$$

Thus, the current step pressure field is required. In order to do so, the divergence operator $\mathbf{M}_1$ is applied to (6.19), provided that $\mathbf{M}_1\mathbf{u}_h^{n+1} = 0$. Thus, the system of equations for the pressure is found (6.20).

$$\mathbf{M}_1\boldsymbol{\Omega}_1^{-1}\mathbf{M}_1^*\mathbf{p}_h = -\frac{1}{\Delta_t}\mathbf{M}_1\mathbf{u}_h^p \tag{6.20}$$

Generally, the relation $-\mathbf{M}_1\boldsymbol{\Omega}_1^{-1}\mathbf{M}_1^*$ is defined as the Laplacian operator $\mathbf{L}_1$ [22], negative-definite and symmetric. Nonetheless, provided that it will be solved using a Conjugate Gradient Method, which requires a positive-definite matrix, the operator will not be defined since it would not be mathematically accurate.

6.1.6 Sample case

In order to test the Symmetry-preserving discretization, a Lid-Driven Cavity was implemented. In this case, nonetheless, the "spanwise" direction had periodic boundary conditions in order to obtain a false bidimensional flow in case of a laminar implementation. Hence, only three control volumes will be distributed along the periodic direction in this analysis.

Thus, the domain simulated may be structured by the following boundary conditions Figure 6.3.

Non-slip boundary condition

In order to preserve the properties in a non-slip boundary condition, a ghost (out-of-domain) control volume had to be added [22]. In this section it will be developed in a two-dimensional case (Figure 6.4). Nonetheless, its extension to a three-dimensional case is straightforward.

In this case, the normal velocity to the wall (dotted arrow) will be directly imposed as the staggered control volume directly appears in the boundary. Thus, in Figure 6.3a, that would be the case for u in the lateral walls (YZ plane) and v for the bottom wall (XZ plane).

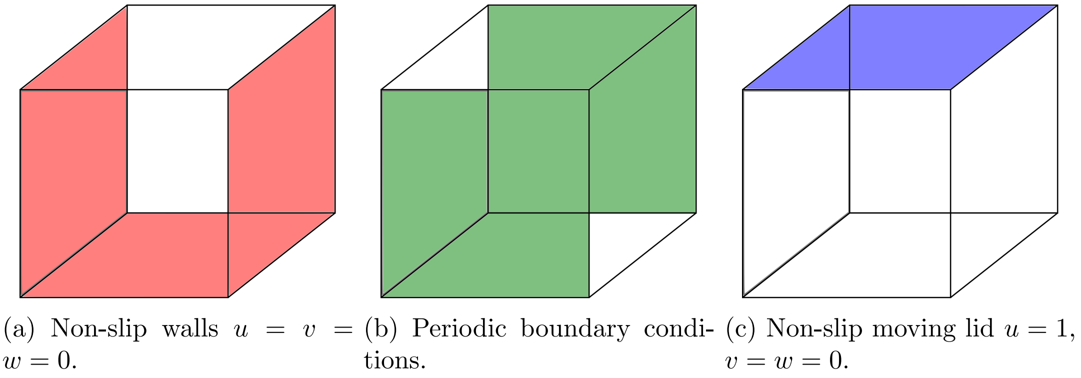

(a) Non-slip walls $u = v = w = 0$.
(b) Periodic boundary conditions.
(c) Non-slip moving lid $u = 1$, $v = w = 0$.

Figure 6.3: Tridimensional boundary conditions.

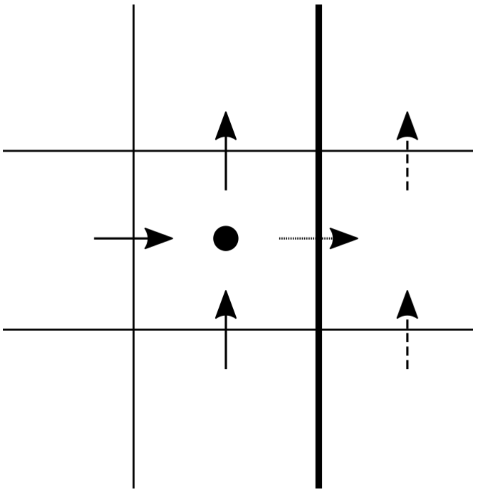


Figure 6.4: Non-slip boundary discretization.

Tangential-to-the-wall velocities, nonetheless, will be indirectly imposed. Let u_b be the velocity required to be imposed in the wall. Then, the same interpolation method applied in the interior nodes (CDS) will be applied at the boundaries considering the ghost velocity u_g, which will appear in the discretization equations in its general form, allowing the code to avoid special cases in the walls when solving transport equations in the boundary nodes. Thus, the ghost velocity will be considered as (6.21) shows. Let $u_{i,b}$ be the velocity corresponding to the node next to the boundary:

$$\begin{aligned} u_b &= \frac{1}{2}(u_g + u_{i,b}) \\ u_g &= 2u_b - u_{i,b} \end{aligned} \tag{6.21}$$

Moreover, no pressure boundary condition will be required, since the construction of the operators directly imposes the pressure behaviour using the divergence operator.

6.1.7 Validation of the symmetry-preserving code

The validation of the symmetry-preserving code will require not just the correct resolution of the Navier-Stokes equations, which will be checked using the classic Lid-Driven Cavity benchmark [6], yet the correct behaviour of the operators will also have to be validated.

In order to do so, given the three main operators $(\mathbf{M}, \mathbf{C}, \mathbf{D})$, three different tests arise, one for each operator.

First of all, the incompressibility of the velocity field has to be ensured. Thus, $\mathbf{M}\mathbf{u}_h = \mathbf{0}$, being this the first of the tests, which will be done repeatedly during the execution of the programme given its dependance on the solution field.

On the other hand, the symmetry of the diffusive operator $\mathbf{D}$ will be validated only at the beginning of the execution provided its dependance on the grid and the Reynolds number. Thus, no update of the velocities will be required to validate this test. In order to do so, $\mathbf{v}^*\mathbf{D}\mathbf{w} = \mathbf{w}^*\mathbf{D}\mathbf{v}$. Thus, the implemented test will verify that

$$\mathbf{v}^*\mathbf{D}\mathbf{w} - \mathbf{w}^*\mathbf{D}\mathbf{v} = 0$$

Eventually, the skew-symmetry of the convective operator will also be validated using an equivalent method. In this case, the relation that has to be verified corresponds to $\mathbf{v}^*\mathbf{C}\mathbf{w} = -\mathbf{w}^*\mathbf{C}\mathbf{v}$. Thus, the implemented test will verify that

$$\mathbf{v}^*\mathbf{C}\mathbf{w} + \mathbf{w}^*\mathbf{C}\mathbf{v} = 0$$

Steady-state solution

In order to determine a steady-state solution a stopping criterion has to be determined. In this case, this criterion will correspond to the timestep-modified L2-norm of the variation of the velocities in every direction.

Let δ_τ be the pondered variation of the velocities:

$$\delta_\tau = \frac{\sqrt{\sum_{\forall i,j,k} (u^{n+1}_{i,j,k} - u^n_{i,j,k})^2 + \sum_{\forall i,j,k} (v^{n+1}_{i,j,k} - v^n_{i,j,k})^2 + \sum_{\forall i,j,k} (w^{n+1}_{i,j,k} - w^n_{i,j,k})^2}}{\Delta_t} \tag{6.22}$$

Solution validation

Using Ghia's benchmark, the solution of the equations has been verified. In order to do so, the norm of the difference of the values at the centerline has been computed using a MATLAB© script which linearly interpolated the results from the output positions to the benchmark positions, allowing the comparison process.

Let u_g and v_g be the velocities from the benchmark [6], n_g the number of points from the benchmark, and $u_{s,g}$ and $v_{s,g}$ the velocities from the output file interpolated onto the benchmark points. Then, the error corresponds to (6.23), being equivalent for v.

$$\varepsilon_u = \sqrt{\sum_{i=1}^{n_g} \left(\frac{u_{g_i} - u_{s,g_i}}{u_{g_i}} \right)^2} \tag{6.23}$$

Thus, for Re = 100, the error is summarized in Table 6.1, yielding the convergence plot for Re = 100 in Figure 6.5.

N	ε_u	ε_v	Computation time [s]
300	0.7138	2.3580	0.1050
1200	0.4823	0.1453	0.8890
2700	0.2490	0.0971	4.8057
4800	0.1829	0.0758	16.3375
7500	0.1475	0.0628	42.4443
10800	0.1207	0.0542	99.8913
24300	0.0790	0.0409	554.8682
30000	0.0700	0.0385	892.9747
36300	0.0627	0.0364	1338.3020
120000	0.0322	0.0277	19477.7700

Table 6.1: Relative error for both u in the vertical midplane $(0.5, y, z)$ and v in the horizontal midplane $(x, 0.5, z)$ for Re = 100 in comparison to [6] with $N_z = 3$.

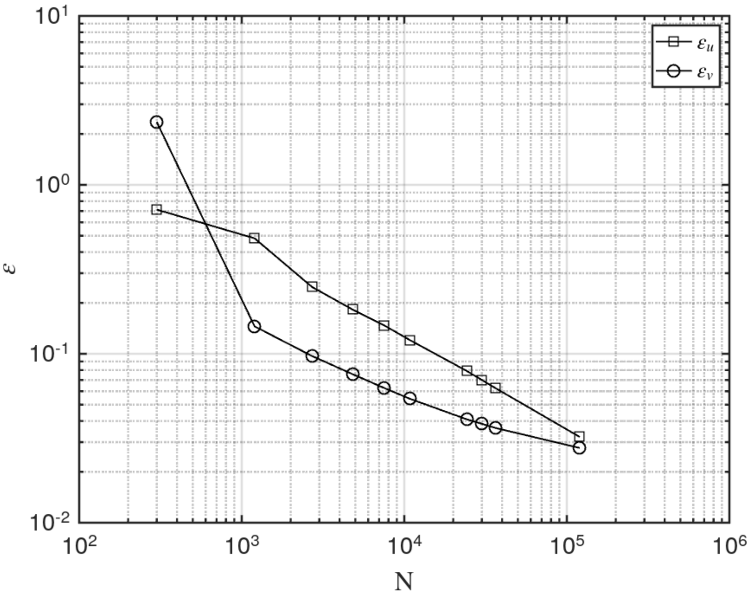


Figure 6.5: Convergence plot for Re = 100 using (6.23).

Thus, in the finer case, the solution yield the velocity distribution from Figure 6.6.

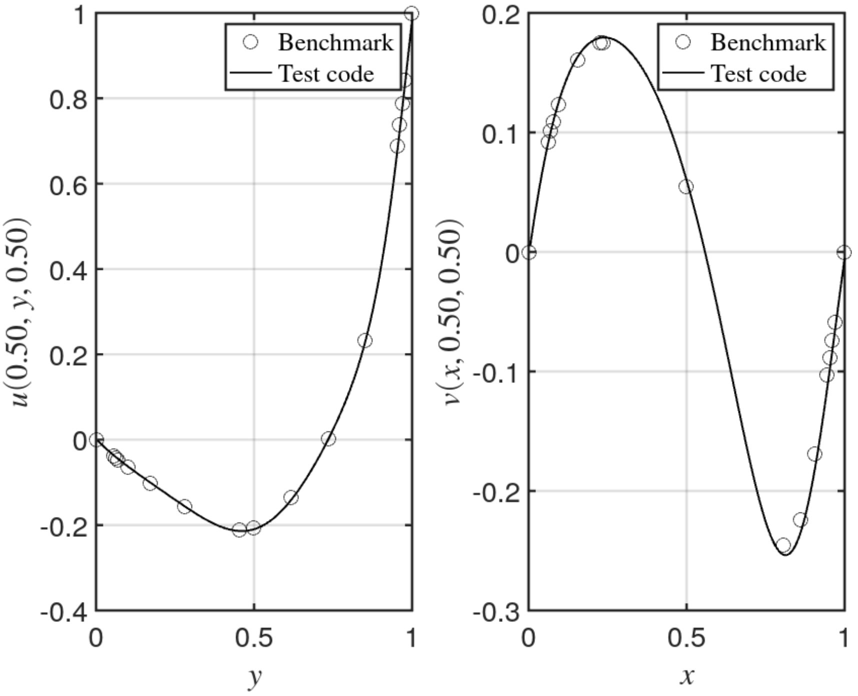


Figure 6.6: Velocities distribution (u, left; v, right) in a Lid Driven Cavity along the lines $y = 0.5; z = 0.5$ and $x = 0.5; z = 0.5$, respectively, of a Re = 100 simulation using 120000 control volumes. The results are compared to the benchmark [6].

To ensure the validity of the solution, it was checked for a second Reynolds (Re = 400), by doing the same procedure as in the last case, yielding its convergence table (Table 6.2) and plot(Figure 6.7).

N	ε_u	ε_v
300	0.8689	0.6083
1200	0.5331	0.3442
2700	0.3656	0.1942
4800	0.2681	0.1411
7500	0.2373	0.1118
10800	0.1864	0.0939
24300	0.1304	0.0641
30000	0.1124	0.0581
36300	0.0982	0.0538
120000	0.0544	0.0349

Table 6.2: Relative error for both u in the vertical midplane $(0.5, y, z)$ and v in the horizontal midplane $(x, 0.5, z)$ for Re = 400 in comparison to [6] with $N_z = 3$.

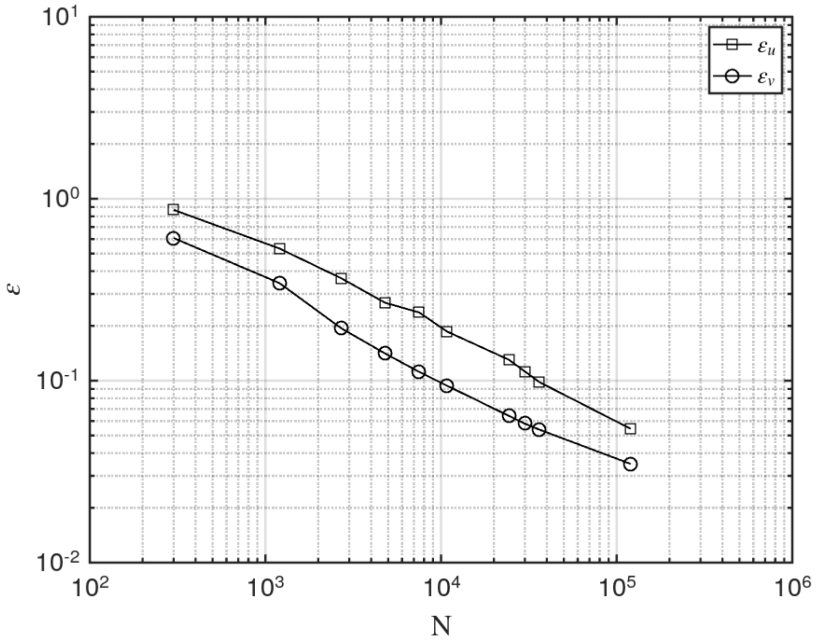


Figure 6.7: Convergence plot for Re = 400 using (6.23).

Mass conservation

The mass conservation test was ran for all inner control volumes where the pressure was solved. The test consisted on verifying the conservation of mass in each and all of them by applying (6.24).

$$(Mu)_{i,j,k} = \bar{u}_{i,j,k} + \bar{v}_{i,j,k} + \bar{w}_{i,j,k} - \bar{u}_{i-1,j,k} - \bar{v}_{i,j-1,k} - \bar{w}_{i,j,k-1} \tag{6.24}$$

Thus, each and every value should be 0. In order to validate the result, the L2-norm (6.25) of the vector was calculated, which had to be equal to zero.

$$|Mu| = \sqrt{\sum_{\forall i,j,k} (Mu)^2_{i,j,k}} \tag{6.25}$$

In this case, the test provided the best results $\mathcal{O}(10^{-11})$ the finer the grid, even though a similar order of magnitude was obtained in all situations.

Symmetry of the diffusive term

In order to do so, the diffusive operator was distributed as $\mathbf{D}_u, \mathbf{D}_v, \mathbf{D}_w$. Then, the test vectors, provided that each operator contribution just affects the corresponding direction of the field, were also splitted using the same methodology. Thus, the actual operation applied was

$$\mathbf{v}_u^*\mathbf{D}_u\mathbf{w}_u + \mathbf{v}_v^*\mathbf{D}_v\mathbf{w}_v + \mathbf{v}_w^*\mathbf{D}_w\mathbf{w}_w - \mathbf{w}_u^*\mathbf{D}_u\mathbf{v}_u - \mathbf{w}_v^*\mathbf{D}_v\mathbf{v}_v - \mathbf{w}_w^*\mathbf{D}_w\mathbf{v}_w$$

which produced an exact 0.0.

Given that the diffusive operator has no dependance on the velocity field, this test was just performed at the beginning of the program execution.

Skew-symmetry of the convective term

In order to test the skew-symmetry of the convective term, a vector fulfilling the boundary conditions was required. Thus, the same velocity vector $\mathbf{u}_h$ was used.

Moreover, the same exact procedure was developed as for the diffusive term, by splitting the full matrix in its three main parts, as well as each vector (6.26).

$$\mathbf{u}_h^* \mathbf{C} \mathbf{u}_h + \mathbf{u}_h^* \mathbf{C} \mathbf{u}_h = 0 \tag{6.26}$$

By doing so, the result obtained in the finest grid for Re = 100, for the first 100 iterations is displayed in Figure 6.8, with an average absolute value of 3.3527×10^{-12}.

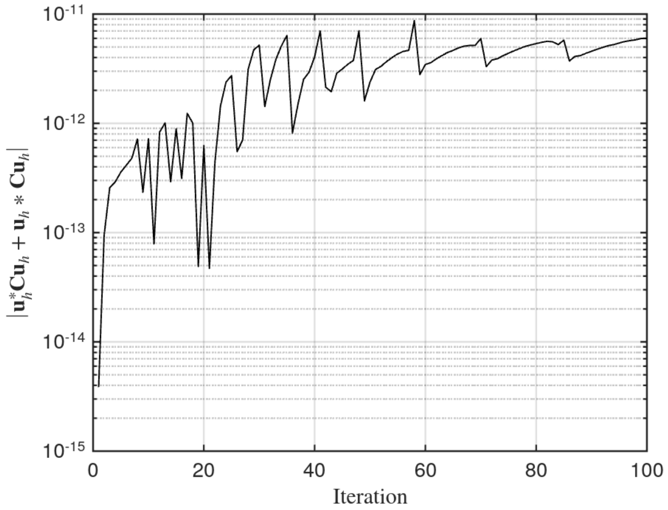


Figure 6.8: Evolution of the result of the convective skew-symmetry test for 120000 control volumes in a Re = 100 Lid-Driven Cavity case.

6.2 LES implementation

The first consideration when implementing a Large-Eddy Simulation program is to choose a proper eddy viscosity model. Hence, it is very important to consider a tradeoff between the computational cost provided that the simulations are performed using 1 CPU and the near-wall behaviour of each model Table 5.1.

Thus, provided its simplicity and a rather low computational cost as well as a $\mathcal{O}(y^3)$ near-wall behaviour, a S3PQ model [18] was selected over other complex models such as WALE. A simple model such as Verstappen's was also programmed to compare the performance of both models. Moreover, due to the simplicity of every model implementation and the fact that all invariants are computed, Vreman's model was also implemented.

Eventually, the improper behaviour of Smagorinksy's model in these kind of wall-bounded flows is alse discussed and compared to the other, proper, LES models.

For the sake of simplicity, in this and further sections concerning the implementation of a Large-Eddy Simulation technique, u_i will be replacing $\tilde{u}_i$.

6.2.1 Invariant computation

The basic element to compute the invariant corresponds to the velocity gradient tensor $G_{ij} = \partial_{x_j} u_i$. Thus, its computation in each control volume has to be discussed.

In order to do so, the method will be developed in a 2D scenario, given that the 3D expansion is straightforward.

Thus, for any given control volume (Figure 6.9), velocities can be interpolated at the desired points so as to proceed with the gradient (6.27), where $v_{i+1/2,j-1/2} = (v_{i,j} + v_{i,j-1} + v_{i+1,j} + v_{i+1,j-1})/4$, being an identical procedure for the other interpolated velocities.

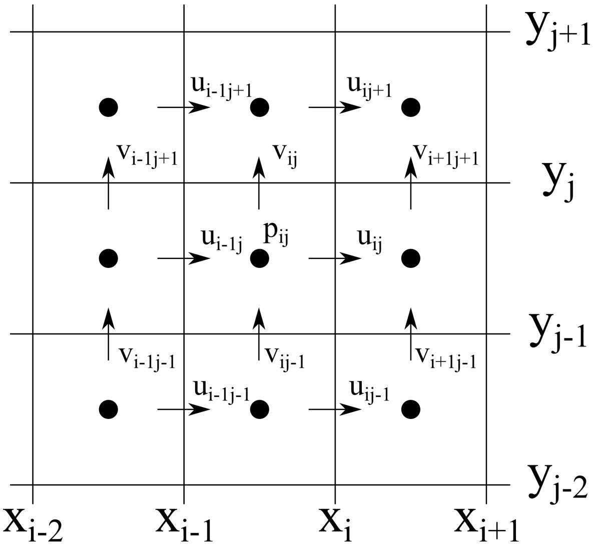


Figure 6.9: Outline of the mesh for the derivation of G_{ij}.

$$\frac{\partial u}{\partial x} = \frac{u_{i,j} - u_{i-1,j}}{x_i - x_{i-1}} \tag{6.27a}$$

$$\frac{\partial u}{\partial y} = \frac{u_{i-1/2,j+1/2} - u_{i-1/2,j-1/2}}{y_j - y_{j-1}} \tag{6.27b}$$

$$\frac{\partial v}{\partial x} = \frac{v_{i+1/2,j-1/2} - v_{i-1/2,j-1/2}}{x_i - x_{i-1}} \tag{6.27c}$$

$$\frac{\partial v}{\partial y} = \frac{v_{i,j} - v_{i,j-1}}{y_j - y_{j-1}} \tag{6.27d}$$

Once the gradient is computed, the computation of each and every invariant requires simple algebra and, thus, is straightforward.

6.2.2 Validation of the LES implementation

Verification of the implementation

In order to verify the model implementation, the wall behaviour of the calculations has been computed. Thus, provided that the S3PQ model's wall behaviour is $\mathcal{O}(y^3)$ (Table 5.1), near the walls, this behaviour has to be followed.

In order to verify this result, the results at the line $x = 0.5,\ z = 0.5$ at the plane with normal $(1, 0, 0)$ have been extracted. The behaviour has been checked for a 900k control volumes (100x100x90) in the top and bottom sections of the plane, yielding an error to a cubic behaviour of 2.8416×10^{-10} (right) and 3.4370×10^{-11} (left), respectively Figure 6.10.

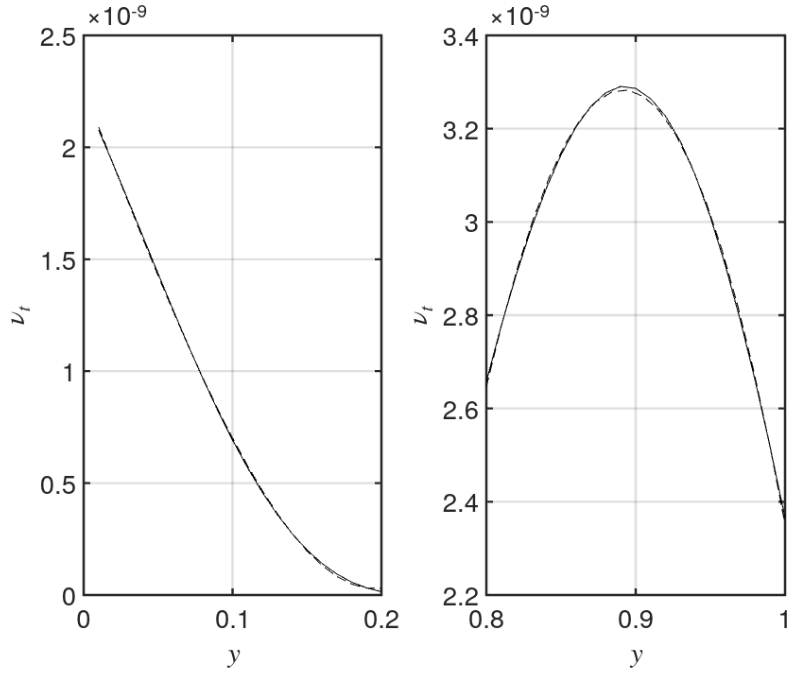


Figure 6.10: Eddy viscosity behaviour in the near wall (top wall, right; bottom wall, left) at the line $x = 0.5,\ z = 0.5$ at the plane with normal $(1, 0, 0)$. The straight line represents the calculated eddy-viscosity, while the dashed line corresponds to the cubic interpolation of the results.

Inaction under laminar conditions

First of all, it is very important to verify that the model does not interfere with the resolution of laminar flows. Thus, for Re = 100, all three models have been tested in a coarse grid (40x40 in plane XY) and in a fine grid (100x100 in plane XY), with the discretization in the z direction adapted to the skewness of the grid.

In order to do so, the results (Table 6.3) have been compared to the benchmark [6] as well as to the DNS solution previously developed.

	Coarse mesh (40x40)				Fine mesh (100x100)			
Model	ε_u	ε_v	$\varepsilon_{DNS,u}$	$\varepsilon_{DNS,v}$	ε_u	ε_v	$\varepsilon_{DNS,u}$	$\varepsilon_{DNS,v}$
No model (DNS)	0.1829	0.0758	-	-	0.0700	0.0385	-	-
S3PQ	0.1873	0.0763	0.0240	0.0066	0.0713	0.0382	0.0185	0.0078
Verstappen	0.1829	0.0759	0.0000	0.0013	0.0701	0.0385	0.0014	0.0000
Vreman	0.2177	0.0981	0.1903	0.2942	0.0826	0.0429	0.1800	0.1143
Smagorinsky	0.1249	0.1153	0.3171	0.5211	0.0491	0.0563	0.2986	0.4623

Table 6.3: Comparison of the error with Ghia of the DNS solution (Table 6.1) as well as the LES cases in a laminar condition.

In the coarse mesh, just smaller than Kolmogorov's lengthscale ($\Delta_x = \Delta_y = 0.025$; $\eta = 0.0316$) in the predominant dimensions, the behaviour of S3PQ and Verstappen's model is excellent, with errors of 2.40% and 0.66% the first and 0.00% and 0.13% the latter when comparing these results with the obtained DNS solution. The results obtained by Verstappen's model are surprisingly good recalling that its near-wall behaviour follows $\mathcal{O}(y^1)$. This might be explained, though, by the small gradient present at the walls in the set laminar conditions, which allows a linear model to perform correctly.

This behaviour, however, can be clearly seen in Vreman's model, having an important leap in the magnitude of the error (19.03% for u and 29.42% for v), which can be explained by this lower near-wall resolution. Nonetheless, the model with a worse behaviour in comparison to the DNS corresponds to the classical Smagorinsky's model, which differs from the DNS in a 31.71% for u and 52.11% for v, having by an important margin the biggest difference when compared to the DNS results.

In regards to the fine mesh, the behaviour is exactly the same as for the coarse, with a general decrease of the difference between the modelled simulations and the DNS.

6.3 Turbulent simulation of a Lid-Driven Cavity

After validating the LES code, a Lid-Driven Cavity turbulent simulation was performed. In order to do so, a Re = 10000 case was considered, even though its simulation until a statistically steady-state was not possible due to the sake of the code, which was serial and implied a much longer, and inaccessible, computing time. Thus, the simulation was performed up to $t = 30$.

The LES model used corresponds to the S3PQ previously used in the validation, and the discretization scale corresponded to $\Delta_x = \Delta_y = \Delta_z \approx 8\eta$. This implies that the control volumes are, approximately, 512 times bigger than Kolmogorov's scale.

The control volumes used were exactly cubical, provided that some problems appeared in the LES solution when more skew control volumes were used.

Even though for a 2D frameset, [6] a steady state solution is proposed at Re = 10000, considering a tridimensional case turbulent phenomena appears mainly in the third dimension. In fact, at $t = 30$, with just the turbulent phenomena starting to appear, that several fluctuations appear in all three velocity components (see Figure 6.11).

In fact, the standard deviation of these values correspond to a 1.59% for u, 4.63% for v and 1.75% for w. On the other hand, in a non turbulent condition, such as Re = 100 these values get reduced to 6.734×10^{-14}%, 4.4289×10^{-6}% and 1.3229×10^{-6}%, respectively.

A much finer mesh would have been optimal, yet the laptop used processing capacity with just a processor made it unfeasible.

This is due to the fact that for Re = 10000, provided the results obtained at $t = 30$, $y^+/y = 858.24$. Thus, each Δ_y corresponded to $\Delta_{y^+} = 7.152$. Thus, the viscous sublayer has not been solved in this simulation at the north wall, but the buffer layer has been simulated with kind of enough good detail.

Up to $t = 10$, the variable evolution was the following for u (Figure 6.12), v (Figure 6.13) and p (Figure 6.14). It is very clear the formation of the main eddy, which, due to the turbulent regime of the fluid, has an irregular shape in terms of u and v. Nevertheless, in terms of pressure, the depression is almost perfectly cylindrical - reminding that the case is three-dimensional - with the exception of Figure 6.14a, which, due to the low development of the case, the pressure just depends on the movement of the lid, thus providing a high pressure zone at the top-right corner of the domain (which appears in all time steps considered) and a low pressure zone at the top-left, which, in fact, will still be there for the remainder of the simulation yet its importance decreases with the appearance of the main eddy.

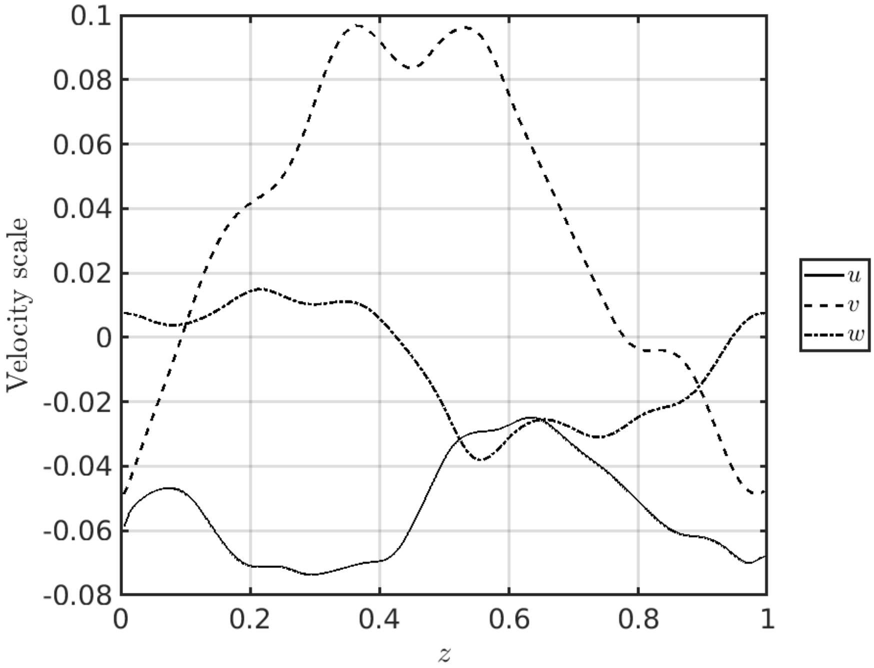
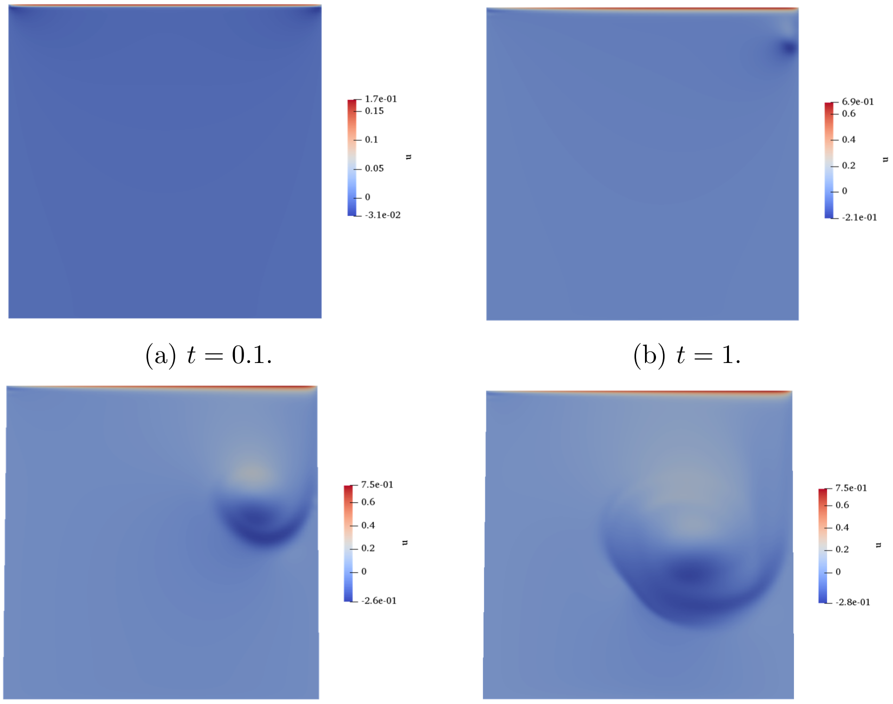


Figure 6.11: Velocity spacial fluctuation at the midline $x = y = 0.5$ of a Lid-Driven Cavity at $t = 30$ with Re $= 10000$ for $\Delta_x = \Delta_y = \Delta_z = 1/120$ using a S3PQ LES model.



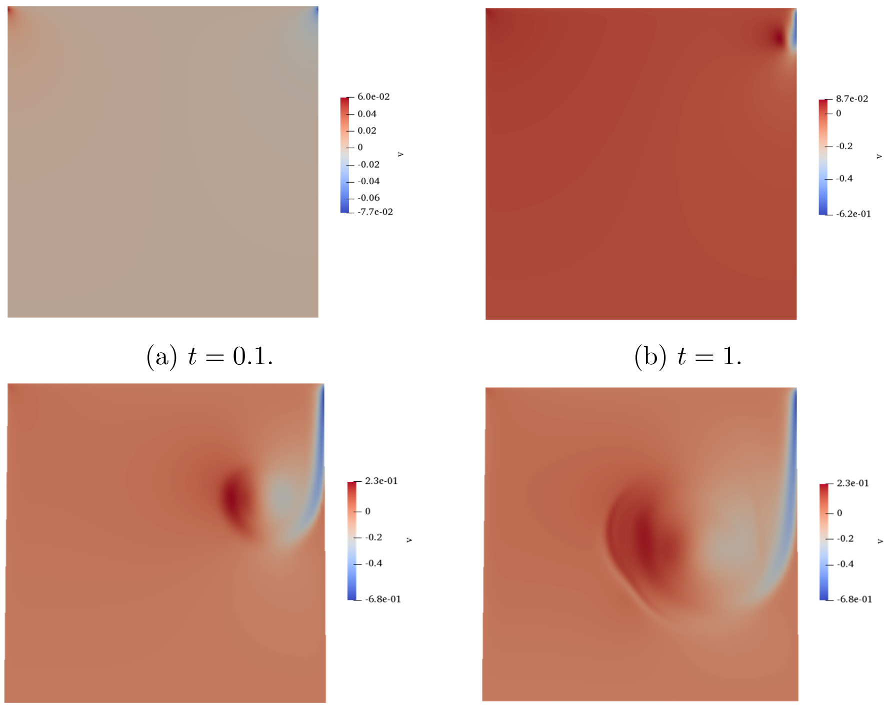
(a) $t = 0.1$.

(b) $t = 1$.

(c) $t = 5$.

(d) $t = 10$.

Figure 6.12: u distribution in plane XY and $z = 0.5$ of Re $= 10000$ simulation under a S3PQ model and $\Delta_x = \Delta_y = \Delta_z = 1/120$ up to $t = 10$.



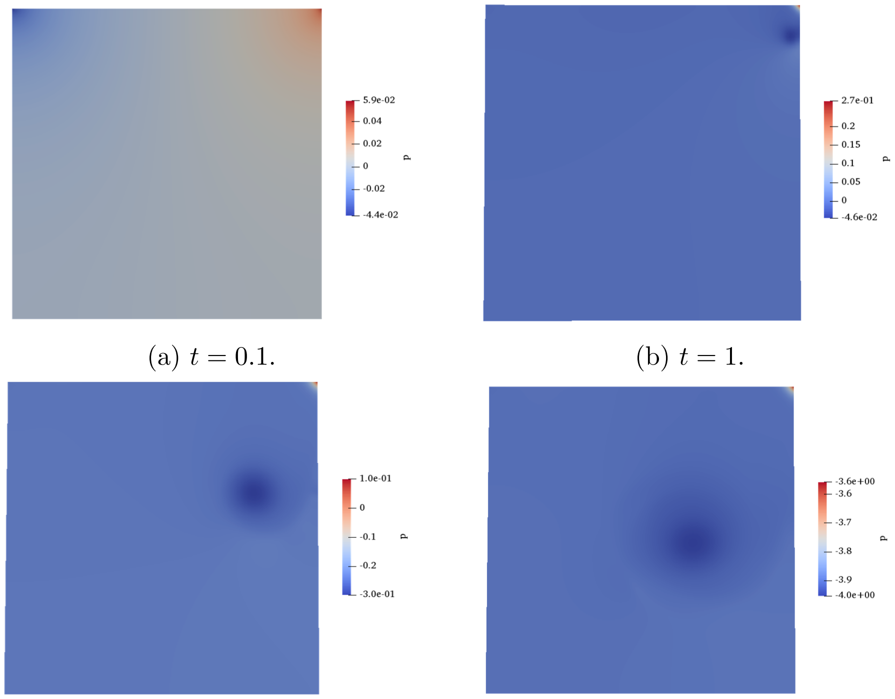
(a) $t = 0.1$.

(b) $t = 1$.

(c) $t = 5$.

(d) $t = 10$.

Figure 6.13: v distribution in plane XY and $z = 0.5$ of Re $= 10000$ simulation under a S3PQ model and $\Delta_x = \Delta_y = \Delta_z = 1/120$ up to $t = 10$.



(a) $t = 0.1$.

(b) $t = 1$.

(c) $t = 5$.

(d) $t = 10$.

Figure 6.14: p distribution in plane XY and $z = 0.5$ of Re $= 10000$ simulation under a S3PQ model and $\Delta_x = \Delta_y = \Delta_z = 1/120$ up to $t = 10$.

One of the interesting points in simulating turbulent flows is analyzing the boundary layer generated by the main flow, and comparing the behaviour to the theoretical approach. Thus, the wall units were computed in this case for two reasons:

1. Validating that the mesh density is enough to simulate all boundary layer layers.
2. Comparing the results to the theoretical approach.

Since the previous approach did not fulfill the first point, with any point inside the viscous sublayer, a denser mesh had to be used. In order to do so, a 240x240x50 mesh was used (2.88M control volumes), which, known the approximate y^+/y ratio previously determined, should imply that at least a node is placed inside the layer itself ($\Delta_y = 0.0042$, being approximately 4η).

Hence, the wall stress $\tau_w = \frac{1}{\mathrm{Re}}\frac{\partial u}{\partial n}$ had to be computed. In order to do so, a MATLAB algorithm was used to obtain a cubic approach of $u = f(y)$ in the near wall region and then easily computing the derivative.

Once obtained the derivative, the frictional velocity was easily obtained as $u_\tau = \sqrt{\tau_w}$. Nonetheless, in order to compute u^+ to behave as expected, the computation changed to

$$u^+ = \frac{1-u}{u_\tau}$$

A similar approach was taken for y^+, yielding Figure 6.15

$$y^+ = \frac{1-y}{\delta_\nu} = (1-y)\mathrm{Re}u_\tau$$

y^+	u^+
1.6966	1.6654
5.0989	4.6878
8.4982	7.0698
11.8974	8.7704
15.2967	9.8969
18.696	10.5995
22.0952	11.0159
25.4944	11.2524

Table 6.4: Results of the simulation at $t = 10$ in regards to the wall units in the north wall of a Lid-Driven Cavity at $\mathrm{Re} = 10000$

As detailed in chapter 4, three different zones are defined: the viscous sublayer, characterized by $u^+ = y^+$; the buffer layer, and the outlaw layer.

In regards to the viscous sublayer, as Table 6.4 shows, the node clearly inside the viscous sublayer ($y^+ = 1.6966$) has an error of 0.01838, whereas the node placed in the boundary between the viscous sublayer and the buffer layer has a bigger error in 0.08063. This may be caused by the lack of points inside the corresponding layer, which could build, with more detail, the behaviour of the flow in this near wall region.

Moreover, provided that the simulating time ($t = 10$) was extremely short given the computational performance of the laptop used, a full turbulent boundary layer in the north wall might not be still fully developed. Thus, the development of a proper buffer layer and outlaw layer, following the theoretical approach to turbulent boundary layers, was not complete, being this the reason why its similarity to the theoretical shape (Figure 4.6) is mere coincidence.

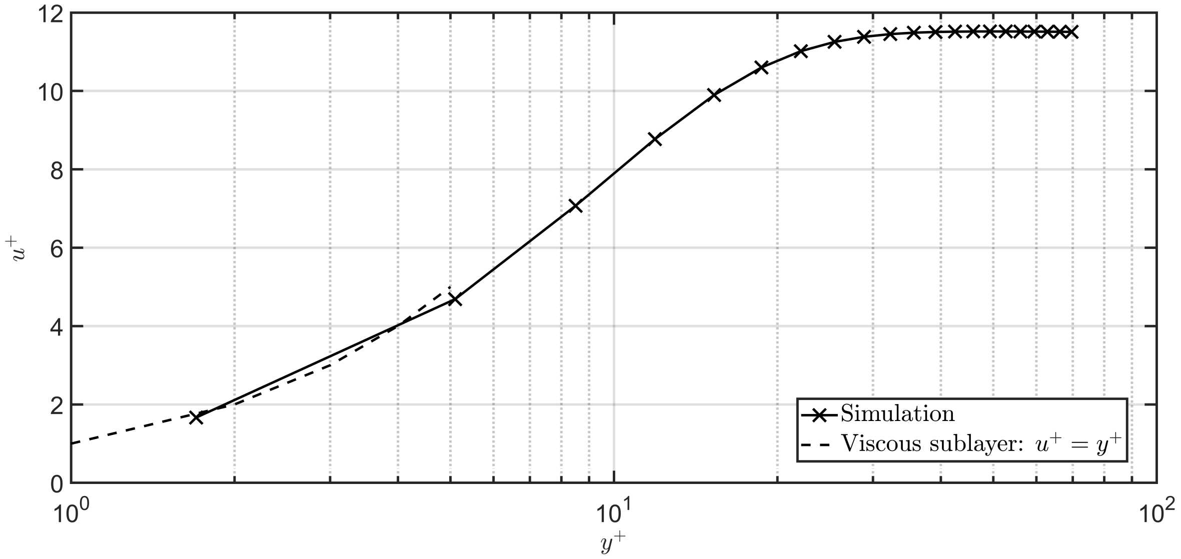


Figure 6.15: $u^+ = f(y^+)$ in a Lid-Driven Cavity north wall following $x = 0.49; z = 0.49$ at $\mathrm{Re} = 10000$ using Verstappen's model at $t = 5$, with $\Delta_x = \Delta_y = 1/200, \Delta_z = 1/50$.

Chapter 7

Conclusions and future research

7.1 Approach to turbulent flows

As stated in the Introduction of this thesis, one of the main goals of this thesis was understanding and getting familiar to how turbulence works and how can be modelled to be simulated. Thus, in chapter 4 as well as in chapter 5 this objective is achieved.

First of all, chapter 4 develops the theoretical general approach to turbulence, in which both the approaches of Reynolds with the mean-flow equations as well as Kolmogorov's, more from a statistical point of view in regards to the energy distribution among all possible lengthscales. Moreover, a small simulation of the Burgers' equation is also developed, which shows, in a simple manner, the energy cascade predicted by Kolmogorov.

Moreover, one of the most interesting features of turbulent flows, the boundary layer, has been deeply studied, firstly considering Prandtl's laminar approach to then put more emphasis on the turbulent boundary layer, in which the wall units play a signficant role, thus developing its approach in the corresponding chapter.

On the other hand, chapter 5 develops the theoretical approach to the Large Eddy Simulation method as well as its application to the Navier-Stokes Equations, being this the case of interest of this thesis. Thus, after developing the eddy-viscosity assumption, the classic Smagorinsky model is developed and its own pros and cons are discussed. This discussion yields that the model is not suitable for the near-wall region. Thus, a different framework approached by some invariants of the tensor divergence of velocity is developed, with some models considered and compared to the classic Smagorinsky model.

Furthermore, as done in the classical approach, a Burgers' equation is also solved by applying to the same framework a spectral LES model such that the behaviour is accurately modelled by using a 20% of the wavenumbers used in the classical DNS approach. However, it was very important to tune correctly the model. Thus, the importance of correctly tuning all LES models arose in that approach to the Burgers' equation.

Thus, at this point all the theoretical basis of turbulence was set so that results obtained from simulations could be well interpreted, being this the reason why the first objective of the thesis was achieved.

7.2 Symmetry-preserving discretizations

During the previous work summarized in chapter 3, the cases regarding Navier-Stokes equations were discretized using various methods, without having any objective apart from making it possible to solve the equations using an own code.

Nonetheless, due to the complexity of turbulence, and in regards to wanting to preserve the physical properties of the flow, turbulent flows are more likely to be accurately solved using a discretization such that the properties of the equations' operators are preserved, which implies that the flow properties will not be perturbed by the model itself. These are the symmetry-preserving discretizations, first introduced by [20].

In the original case, Verstappen and Veldman developed a fourth-order approach, yet due to the complexity of the method, the second-order approach was used in this case.

In chapter 6 this approach is developed to a three-dimensional structured grid and its properties are discussed. Moreover, since all cases in chapter 3 were two-dimensional, the method is tested under laminar conditions in a Lid-Driven Cavity, being its results compared to [6].

With this being validated, the implementation of the LES invariant framework detailed in chapter 5 is performed for multiple methods and, eventually, is verified in the near-wall region considering the order of the approach at that region as well as its inaction under laminar conditions. After considering all different approaches, S3PQ and Verstappen models are selected to perform the turbulent simulation, provided the better wall resolution from the first, as well as the simplicity of the latter.

Thus, a turbulent simulation from a flow at Re $=$ 10000 in a Lid-Driven Cavity was performed to ensure that its performance would be accurate in turbulent conditions in the use of a LES model. Nonetheless, and due to the lack of computational power of a laptop, the viscous sublayer was not properly simulated, since the mesh was still too coarse, even using 1.728M control volumes.

Hence, a finer mesh (240x240x50) was used for a simulation up to $t = 10$ under the same exact conditions, which allowed the obtention of some nodal points inside the viscous sublayer as Table 6.4 shows. Moreover, the most inner points had the most similar behaviour to the predictions of the theory, yet, in the outer layers, the flow behaviour was not as expected. This may have happened due to the following reasons: first of all, the mesh might still be too coarse to compute properly the turbulent boundary layer, even though it was fine enough to obtain physically solid results in the nearest points; and, on the other hand, a too short simulating time, which did not allow the turbulent boundary layer to fully develop and, thus, providing not accurate results.

7.3 Future research

7.3.1 High-Performance Computing

Up to this point, all of the codes used perform in just a single CPU, yet the field of CFD is based on a parallel framework of computing: multiple CPUs - and even GPUs - will perform much strongly than a single processing unit. Thus, the extension of the code - mainly the solver, the other parts are quite trivial - to a parallel framework is important.

For this reason, a two-dimensional conjugate gradient solver has been implemented in a parallel framework and tested against other solvers in both serial and parallel frameworks (see Figure 7.1). This has been done in the subject *High Performance Computing in Aerospace Engineering*.

Thus, this parallel code has to be extended to a three-dimensional case, as well as improving the performance of the code by optimizing the domain division and the halo updates such that a division in all dimensions could be performed, and not just a single dimension

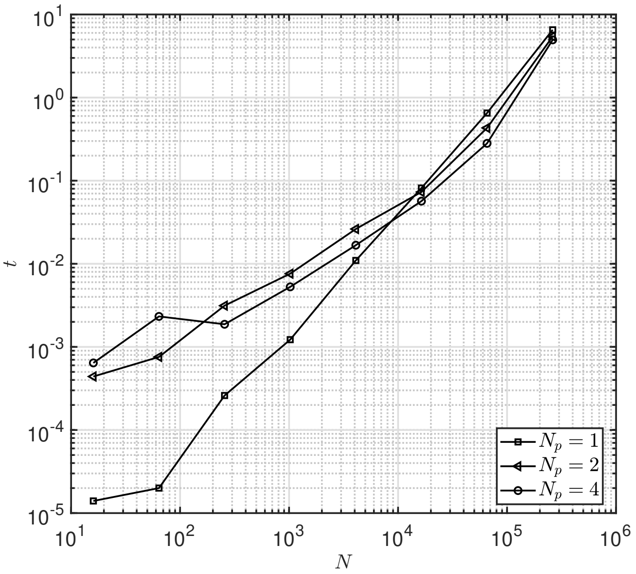


Figure 7.1: Comparison of the results of a serial two-dimensional CG solver with the parallel code for $N_p = 2$ and $N_p = 4$, with $\delta = 10^{-6}$.

division as Figure 7.2 shows, which correspond to the code developed.

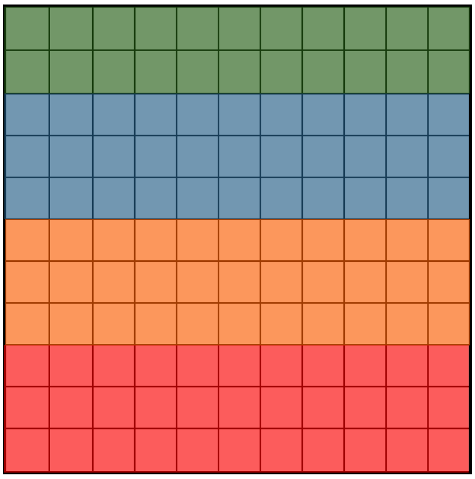


Figure 7.2: Domain division for 4 CPUs in the case of Figure 7.1.

This would allow the computation of denser meshes for extended simulating times in smaller computing times, yielding a more efficient code that would be able to provide the results not reached in this Bachelor's thesis.

7.3.2 More efficient linear solvers

Even though the conjugate gradient used is not an extremely bad linear solver for a big number of equations (5M equations are solved in my laptop in approximately a minute) in a serial code, much more efficient solvers - and relatively easy to parallelize - can be found such as the Multigrid [25], which is based on the residual equation and the use of coarser grids than the original to obtain an accurate yet quick solution to the system of equations. Some cases have been unsuccessfully coded, thus, much work has to be put on this case to obtain an efficient solver.

Another line to work on corresponds to the preconditioning of the conjugate gradient method, such that its performance is boosted. This case has also been unsuccessfully tested during this thesis and thus has not been used, even though its effectiveness is proven and could be effectively used to provide faster solutions to the system of equations.

7.3.3 Better understanding of turbulent phenomena

Even though the development of this thesis has been a mind-opener about the turbulent phenomena, it has not developed an excellent knowledge of the theoretical developments around this phenomena. Thus, in order to successfully progress in this field, a very deep knowledge on the physics of turbulence should be a must. Thus, there is a lot of work to do in this field so as to being able to give a helping hand in the research fields concerning this phenomena.

Bibliography

1. FEYNMAN, Richard P; LEIGHTON, Robert B; SANDS, Matthew. *European Journal of Physics*. Vol. 24, The Feynman Lectures on Physics, Vol. I: The New Millennium Edition: Mainly Mechanics, Radiation, and Heat. 2011. ISBN 9780465024933. ISSN 01430807. Available also from: `http://books.google.at/books?id=bDF-uoUmttU`.
2. POPE, Stephen B. *Turbulent Flows*. 7th editio. New York: Cambridge University Press, 2010. ISBN 978-0-521-59886-9.
3. PATANKAR, Suhas V. *Numerical heat transfer and fluid flow*. Washington : Hemisphere Pub. Corp., 1980. ISBN 0891165223. Available also from: `https://discovery.upc.edu/iii/encore/record/C__Rb1014633__Spatankar__Orightresult__U__X4?lang=cat`.
4. CENTRE TECNOLÒGIC DE TRANSFERÈNCIA DE CALOR (CTTC). *Introduction to the Fractional Step Method by CTTC*. Terrassa. Technical report. Centre Tecnològic de Transferència de Calor.
5. PLANA RIU, Josep. *Comparison between FSM and SIMPLER in the resolution of the Navier-Stokes Equations*. Terrassa, 2019. Technical report. Centre Tecnològic de Transferència de Calor.
6. GHIA, U; GHIA, K N; SHIN, C T. *High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method*. 1982. Technical report.
7. VAHL DAVIS, G de; JONES, I P. Natural convection in a square cavity: a comparison exercise. *International Journal for Numerical Methods in Fluids*. 1983, vol. 3, pp. 227–248.
8. VAHL DAVIS, G. de. Natural convection in a square cavity: a benchmark numerical solution. *International Journal for Numerical Methods in Fluids*. 1983, vol. 3, pp. 249–264.
9. PLANA RIU, Josep. *Resolution of a Differentially Heated Cavity using FSM*. Terrassa, 2019. Technical report. Centre Tecnològic de Transferència de Calor.
10. KUNDU, Pijush K; COHEN, Ira M; AYYASWAMY, P S; HU, Howard H. Turbulence. In: *Fluid Mechanics*. 7th editio. San Diego: Elsevier Science & Technology, 2010, chap. 13, pp. 537–602. ISBN 978-0-123-81399-2.
11. RICHARDSON, Lewis F. *Weather prediction by numerical processes*. 1st. Cambridge: Cambridge University Press, 1922.
12. KOLMOGOROV, Andrei Nikolaevich. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *Proceedings of the Royal Society*. 1991, vol. 434, pp. 9–13.
13. SCHLICHTING, Hermann; GERSTEN, Klaus. *McGraw-Hill*. Boundary-layer Theory. 9th ed. Berlin: Springer-Verlag Berlin Heidelberg, 2017. ISBN 9783662529171. ISSN 00029505. Available from DOI: `10.1119/1.1971812`.

14. GERSTEN, Klaus; HERWIG, Heinz. *Strömungsmechanik. Grundlagen der Impuls-, Wärme- und Stoffübertragung as asymptotisher Sicht.* Wiesbaden: Springer Fachmedien, 1992. ISBN 9783322939715. Available from DOI: `10.1007/978-3-322-93970-8`.

15. SAGAUT, Pierre. *Large Eddy Simulation for Incompressible Flows. An Introduction.* 3rd. Berlin: Springer-Verlag Berlin Heidelberg, 2006. ISBN 978-3-540-26344-9.

16. LESIEUR, M.; YAGLOM, A.; DAVID, F. *Les Houches: École d'été de Physique Theorique 74.* New trends in turbulence. Les Houches: Elsevier Academic Press, 2002. ISBN 9783540429784.

17. SMAGORINSKY, J. General Circulation Experiments With the Primitive Equations. *Monthly Weather Review.* 1963, vol. 91, no. 3, pp. 99–164. ISSN 0027-0644. Available from DOI: `10.1175/1520-0493(1963)091<0099:gcewtp>2.3.co;2`.

18. TRIAS, F. X.; FOLCH, D.; GOROBETS, A.; OLIVA, A. Building proper invariants for eddy-viscosity subgrid-scale models. *Physics of Fluids.* 2015, vol. 27, no. 6. ISSN 10897666. Available from DOI: `10.1063/1.4921817`.

19. NICOUD, F.; DUCROS, F. Subgrid-scale stress modelling based on the square of the velocity. *Flow, Turbulence and Combustion.* 1999, vol. 62, pp. 183–200. ISBN 0021-9991. ISSN 1294-4475. Available from DOI: `10.1016/j.jcp.2004.10.018`.

20. VERSTAPPEN, R. W.C.P.; VELDMAN, A. E.P. Symmetry-preserving discretization of turbulent flow. *Journal of Computational Physics.* 2003, vol. 187, no. 1, pp. 343–368. ISSN 00219991. Available from DOI: `10.1016/S0021-9991(03)00126-8`.

21. TRIAS, F. X. What can operators tell us about Navier-Stokes equations? In: *Turbulence: phenomenology, simulation, aerodynamics.* Terrassa, 2019.

22. TRIAS, F. X.; LEHMKUHL, O.; OLIVA, A.; PÉREZ-SEGARRA, C. D.; VERSTAPPEN, R. W.C.P. Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured grids. *Journal of Computational Physics.* 2014, vol. 258, pp. 246–267. ISSN 10902716. Available from DOI: `10.1016/j.jcp.2013.10.031`.

23. CHORIN, Alexandre Joel. Numerical Solution of the Navier-Stokes Equations. *Mathematics of Computation.* 1968, vol. 22, no. 104, pp. 745–762.

24. TRIAS, F. X.; LEHMKUHL, O. A self-adaptive strategy for the time integration of Navier-Stokes equations. *Numerical Heat Transfer, Part B: Fundamentals.* 2011, vol. 60, no. 2, pp. 116–134. ISSN 10407790. Available from DOI: `10.1080/10407790.2011.594398`.

25. BRIGGS, William L; HENSON, Van Emden; MCCORMICK, Steve F. *A Multigrid Tutorial.* 2nd. Philadelphia: Society for Industrial and Applied Mathematics, 2000. No. December 2015. ISBN 0-89871-462-1.

26. CENTRE TECNOLÒGIC DE TRANSFERÈNCIA DE CALOR (CTTC). *A Two-dimensional Transient Conduction Problem.* Terrassa. Technical report. Centre Tecnològic de Transferència de Calor.

27. INCROPERA, Frank P.; BERGMAN, Theodore L.; LAVINE, Adrienne S.; DEWITT, David P. *Fundamentals of Heat and Mass Transfer.* 7th. Danvers: John Wiley & Sons, 2011. ISBN 978-0470-50197-9.

28. CENTRE TECNOLÒGIC DE TRANSFERÈNCIA DE CALOR (CTTC). *Numerical Solution of Convection.* Terrassa. Technical report. Centre Tecnològic de Transferència de Calor.

29. PLANA RIU, Josep. *Validation of the Convection-Diffusion Equation.* Terrassa, 2019. Technical report. Centre Tecnològic de Transferència de Calor.

30. GREIF, Chen. *Numerical Solution of Linear Systems.* Vancouver B.C., 2008. Available also from: `http://www.cs.tau.ac.il/~dcor/Graphics/adv-slides/Solving.pdf`. Technical report. The University of British Columbia.

31. ÇENGEL, Yunus A; CIMBALA, John M. *Fluid Mechanics: fundamentals and applications.* 4th. New York: McGraw Hill Education, 2018. ISBN 978-1-259-69653-4.

Appendix A

2D Conduction

A.1 Introduction

A.1.1 Problem definition

A very long rod is composed of four different materials (M_1 to M_4), represented with different colours in Figure A.1, where all the lines are parallel to the corresponding coordinate axis, where $p_1 = (0.50, 0.40)\ m$, $p_2 = (0.50, 0.70)\ m$ and $p_3 = (1.10, 0.80)\ m$.

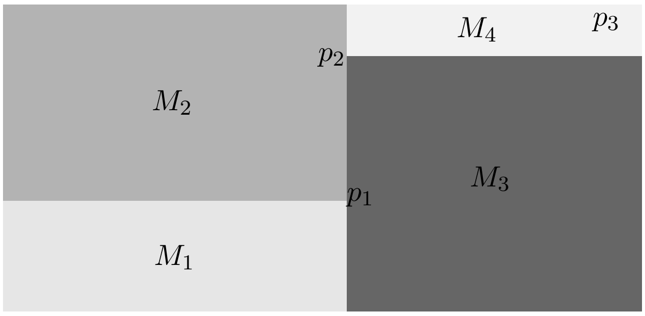


Figure A.1: General outline of the problem's rod. Extracted from [26]

Each material from the rod will have the properties specified in Table A.1, whereas the **boundary conditions** for the rod will be described in Table A.2.

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

Table A.1: Physical properties of the materials. Extracted from [26]

A.2 Governing equations

Since the assignment is about a conduction problem, first of all the theoretical background must be stated. In fact, conduction's background is based around **Fourier's Law** (A.1).

[1] Inlet heat flux

Wall	**Boundary condition**
Bottom	Isotherm at $T = 23.00\ °C$
Top	Uniform $Q_{flow} = 60.00\ W/m$ [1]
Left	In contact with a fluid at $T_g = 33.00\ °C$ and a heat transfer coefficient $h = 9.00\ W/m^2K$
Right	Uniform temperature $T = 8.00 + 0.005t\ °C$, where t is the time in seconds

Table A.2: Boundary conditions. Extracted from [26]

$$\mathbf{q} = -\lambda \nabla T \tag{A.1}$$

where λ stands for the **thermal conductivity**, which is characteristic of each material.

Since the main objective of this assignment, as well as any conduction problem, is determining the **temperature field**. Following the procedure of pp. 82-85 of [27], for a single control volume, the **Heat Diffusion equation** will be obtained (A.2).

$$\frac{\partial}{\partial x_i}\left(\lambda \frac{\partial T}{\partial x_i}\right) + \dot{q}_v = \rho c_p \frac{\partial T}{\partial t} \tag{A.2}$$

A.2.1 Finite Volume Approach

In order to program (A.2), some kind of discretization must be done. In this case, a Finite Volume Method (or Control Volume Method) will be applied. Hence, the equation has to be integrated for two dimensions in our case ($i = 1, 2$ in (A.2)). This process is going to be done following [3] in a one dimensional situation, and then extrapolating the results [2].

From

$$\frac{\partial}{\partial x}\left(\lambda \frac{\partial T}{\partial x}\right) = \rho c_p \frac{\partial T}{\partial t} \tag{A.3}$$

Integrating (A.3) within a control volume limited by points w and e (Figure A.2) and over a time interval t to $t + \Delta t$

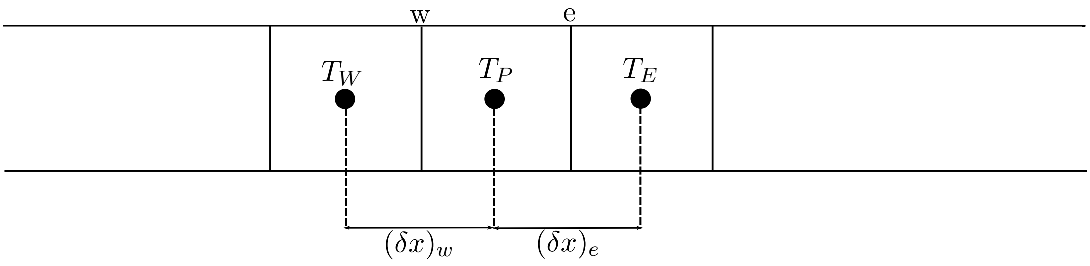


Figure A.2: Control volumes at 1D

This leads to

$$\rho c_p \int_w^e \int_t^{t+\Delta t} \frac{\partial T}{\partial t}\, dt\, dx = \int_w^e \int_t^{t+\Delta_t} \frac{\partial}{\partial x}\left(\lambda \frac{\partial T}{\partial x}\right)\, dt\, dx$$

In this case, where the discretization will be set to be a node-centered scheme, λ_e and λ_w will be considered as the harmonic mean between the λ values at E and P (in the case of λ_e) and P and W (in the case of λ_w). For the first case, with a natural extension to the second one, this mean is defined by (A.4).

[2] Remark that the source is now not considered, it will be introduced further.

$$\lambda_e = \frac{(\delta x)_e}{\frac{\lambda_E}{(\delta x)^+} + \frac{\lambda_P}{(\delta x)^+}} \tag{A.4}$$

Where $\delta x)^+$ and $\delta x)^-$ are the distances from the interface to E and from P to the interface, respectively.

The first member of the equation can be rewritten as $\rho c_p \Delta_x (T_P - T_P^0)$, where T_P^0 stands for the latest temperature known of the corresponding point P.

Now considering that temperature varies linearly between each node, the second term will be expressed as

$$\int_t^{t+\Delta_t} \left[\frac{\lambda_e (T_E - T_P)}{(\delta x)_e} - \frac{\lambda_w (T_P - T_W)}{(\delta x)_w} \right] \, dt$$

Assuming that temperature varies in time following a general expression (A.5) ruled by a weighting factor f between 0 and 1, (A.6) is found.

$$\int_t^{t+\Delta_t} T_P \, dt = [f T_P + (1-f) T_P^0] \, \Delta_t \tag{A.5}$$

$$\rho c_p \frac{\Delta_x}{\Delta_t} (T_P - T_P^0) = f \left[\frac{\lambda_e (T_E - T_P)}{(\delta x)_e} - \frac{\lambda_w (T_P - T_W)}{(\delta x)_w} \right] + (1-f) \left[\frac{\lambda_e (T_E^0 - T_P^0)}{(\delta x)_e} - \frac{\lambda_w (T_P^0 - T_W^0)}{(\delta x)_w} \right] \tag{A.6}$$

where Δ_x is the distance between the two interfaces of a 1D discretization control volume.

Rearranging values, and introducing the source term, which was linearized as $\dot{q} \Delta_x \Delta_t = S_C + S_P T_P$, and considering $f = 1$, which is called a **fully implicit scheme**, (A.7) is derived, what is called the **Fully Implicit Discretization Equation**.[3]

$$a_P T_P = a_E T_E + a_W T_W + b \tag{A.7}$$

where

- $a_E = \frac{\lambda_e}{(\delta x)_e}$
- $a_W = \frac{\lambda_w}{(\delta x)_w}$
- $a_P^0 = \frac{\rho c_p \Delta x}{\Delta t}$
- $b = S_C \Delta x + a_P^0 T_P^0$
- $a_P = a_E + a_W + a_P^0 - S_P \Delta x$

Now, considering a 2D discretization, a north and south terms will appear, as well as some extra terms for the constants regarding the surfaces of the extra dimension control volumes (A.8).

$$a_P T_P = a_E T_E + a_W T_W + a_N T_N + a_S T_S + b \tag{A.8}$$

where

[3] In the case where $f = 0.5$, the scheme is called **Crank-Nicholson**, while at $f = 0$, the scheme is called explicit

- $a_E = \frac{\lambda_e \Delta_y}{(\delta x)_e}$
- $a_W = \frac{\lambda_w \Delta_y}{(\delta x)_w}$
- $a_N = \frac{\lambda_n \Delta_x}{(\delta y)_n}$
- $a_S = \frac{\lambda_s \Delta_x}{(\delta x)_s}$
- $a_P^0 = \frac{\rho c_p \Delta_x \Delta_y}{\Delta_t}$
- $b = S_C \Delta_x \Delta_y + a_P^0 T_P^0$
- $a_P = a_E + a_W + a_P^0 - S_P \Delta_x \Delta_y$

In order to solve, now, this 2D problem numerically, a Gauss-Seidel line by line Method using a TriDiagonal Matrix Algorithm will be used.

A.2.2 TriDiagonal Matrix Algorithm (TDMA)

The **TriDiagonal Matrix Algorithm** is a solver for linear system of equations based on the fact that the coefficient matrix of the correspoding system is a tridiagonal matrix

$$\begin{pmatrix} a_0 & b_0 & & & \\ c_1 & a_1 & b_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots \\ & & & c_N & a_N \end{pmatrix}$$

Then, the system of equations in a conduction system takes the form of (A.9).

$$a_i T_i = b_i T_{i+1} + c_i T_{i-1} + d_i \tag{A.9}$$

By operating algebraically with the fact that a $T_i = P_i T_{i+1} + Q_i$ relation is being seeked, the values of P_i and Q_i can be easily obtained using (A.10) and (A.11).

$$P_i = \frac{b_i}{a_i - c_i P_{i-1}} \tag{A.10}$$

$$Q_i = \frac{d_i + c_i Q_{i-1}}{a_i - c_i P_{i-1}} \tag{A.11}$$

Hence, since it is possible to know the last temperature (in our case is given for all the time domain analysed), once known all the P_i and Q_i values, the temperatures for a 1D problem will be found.

A.2.3 Gauss-Seidel line by line

Now, the 2D domain can be integrated line by line by considering the north and south terms in the independent term of each equation. Hence, by considering

$$a_{ij} T_{ij} = a_{(i+1)j} T_{(i+1)j} + a_{(i-1)j} T_{(i-1)j} + a_{i(j+1)} T_{i(j+1)} + a_{i(j-1)} T_{i(j-1)} + b_{ij}$$

- $a_i = a_{ij}$

- $b_i = a_{(i+1)j}$
- $c_i = a_{(i-1)j}$
- $d_i = a_{i(j+1)}T^*_{i(j+1)} + a_{i(j-1)}T^*_{i(j-1)} + b_{ij}$

where T^* stands for the last iteration temperature in the Gauss-Seidel iterative cycle. Hence, for each j a TDMA can be applied in order to integrate all the 2D Domain.

A.3 Numerical approach to the problem

In order to get the numerical approach to the problem, the rod's section must be discretized. Generally, an input of a numerical problem corresponds to the number of control volumes. Nevertheless, in the corresponding case, due to the fact that the rod has four well defined materials, it is interesting to have the control volumes corresponding to the boundaries of each material, as it can be seen in Figure A.3.

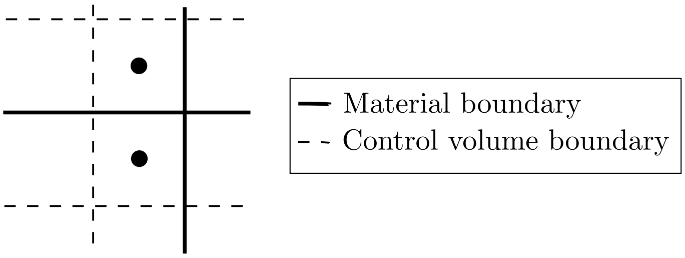


Figure A.3: Control volumes disposition in material boundaries

Hence, not all the number of control volumes will be the easier way to obtain a good solution of the problem, since equilibrium for control volumes will be easier to determine. In order to do so, the lengths for the x and y direction can be normalized, allowing the problem to have the discretization applied.

Since in x direction, the length corresponds to 1.1 m, by normalizing the distances with this maximum length the fact that the distance must be divided by a multiple of 11 is obtained.

Following the same reasoning, in y direction the distance must be divided by a multiple of 8.

In order to do so, once chosen the number of control volumes of the solver, this will give the steps in the x and y direction, starting at $(0.0, 0.0)$ m and finishing at $(1.1, 0.8)$ m. Hence, the mid position in x and y directions of each and every control volume will correspond to the node.

A.3.1 Boundary conditions

Once discretized the space studied, the boundary conditions have to be applied. Due to the fact that all the boundary conditions are different, every part of the boundary has to be taken in account differently.

Left boundary condition

On the left side of the rod, a fluid with $h = 9.00\ W/m^2K$ at $T_g = 33.00\ °C$ keeps contact with the left wall. By convection (A.12) rises. Hence, the equilibrium in the boundary would be described by $a_P^* T_B^{(L)} = a_1 T_1 + h(T_g - T_B^{(L)})\Delta x \Rightarrow (a_P^* + h\Delta_x)T_B^{(L)} = a_1 T_1 + \Delta_x h T_g$, as it can be seen in Figure A.4.

$$q_B = h(T_g - T_B) \tag{A.12}$$

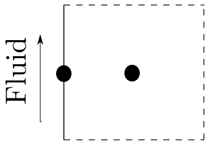


Figure A.4: Left Boundary Condition applied onto a discretized domain

Right boundary condition

.

On the right side, a condition of $T = 8.00 + 0.005t\ °C$ is applied. This is the simplest condition that can be given, since the temperature at the boundary will not have to be calculated, as it would happen in a heat flux or a fluid. Then, $T_B^{(R)}$ is uniform in all the distribution of y, but varies in time.

Top boundary condition

In this case, an inlet heat flux of $q = 60\ W/m^2$ will be considered. Hence, as it was done in the left boundary condition, the heat equilibrium must be applied here in order to consider the boundary condition. Hence, using the discretized equations in a 1D domain (y, vertical):

$$a_P T_B^{(T)} = a_S T_S + q\Delta_y \tag{A.13}$$

Nevertheless, since Gauss-Seidel will be applied in horizontal lines, the vertical variations will not be considered directly, what forces the substitution of the top boundary condition at the equations when j corresponds to the last of free nodes ((A.13)).

Beginning from

$$a_{ij}T_{ij} = a_{(i+1)j}T_{(i+1)j} + a_{(i-1)j}T_{(i-1)j} + a_{i(j+1)}T_{i(j+1)} + a_{i(j-1)}T_{i(j-1)} + b_{ij}$$

If j is the last value, $j+1$ terms will correspond to the top boundary condition. Hence, since $T_B^{(T)}$ can be obtained from (A.13), where now $T_S = T_{ij}$, and $a_P = a_S = \frac{\lambda_s}{(\delta y)_s}$. (A.13) can be modified to obtain $T_B^{(T)} = T_{ij} + \frac{q\Delta_y}{a_P} = T_{ij} + q_{top}$. Now, substituting in the general equation and operating algebraically, the new TDMA equation for this condition will be (A.14).

$$(a_{ij} - a_{i(j+1)})T_{ij} = a_{(i+1)j}T_{(i+1)j} + a_{(i-1)j}T_{(i-1)j} + a_{i(j-1)}T^*_{i(j-1)} + b_{ij} - a_{i(j+1)}q_{top} \tag{A.14}$$

Bottom boundary condition

Finally, the bottom boundary condition corresponds to a uniform and constant temperature of 23.00 $°C$. Its application correspond to the right boundary condition (See section A.3.1).

Implementation of the boundary conditions

The boundary conditions will be implemented taking in account bounded nodes, which can be studied by considering two different options:

- Half control volumes
- Infinitesimal control volumes

In the case concerning this report, the second option will be chosen, as it can be seen in Figure A.4, for instance. This will give a discretization corresponding to Figure A.5.

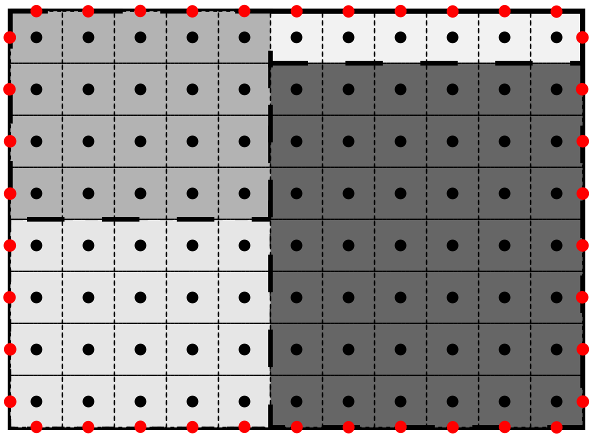


Figure A.5: Discretization with bounded (red) and free (black) nodes

A.4 Algorithm

In this section, the algorithm used will be summarized.

1. Introduce the geometrical, material, physical properties and numerical conditions for the problem solver.
2. Define the mesh
3. Calculate the time independent coefficients $a_N, a_S, a_E, a_W, a_P, a_P^0$
4. Solve the problem. For each timestep:
   (a) Calculate the b coefficient.
   (b) Update $T_R^{(boundary)}$
   (c) Process Gauss-Seidel line by line.
      i. For each line, use TDMA considering the last temperature field is solution.
      ii. Calculate the norm of equation

$$a_P T_P - a_N T_N - a_S T_S - a_E T_E - a_W T_W - b = 0$$

      iii. If tolerance is achieved, next time step. Else, restart the process of Gauss-Seidel.
   (d) Store definitive values in corresponding matrices
5. Print results

A.5 Results obtention

Once the code is written using the Finite Volume Method (FVM), the results demanded must be obtained. In order to do that, by considering the fact that the points asked are not exactly any of the points in the mesh, an interpolation must be done. In this case, the method used is going to be a **bilinear interpolation**, since the case corresponds to a 2D scalar field, with small difference in between the grid points. Furthermore, temperature has been considered to vary linearly in time. Hence, considering a spatial linear variation can be considered.

Once this is implemented, the code was used with different mesh sizes in order to evaluate the variation in the results. Once this analysis is done, the temperature is set to keep getting constant to a value.

As it is developed in subsection A.6.3, a smaller mesh size tends to a constant value as a result of the numerical simulation. Nevertheless, what also has to be taken in account is the fact that for a smaller mesh size, the computational time will be higher. Therefore, it is important not just to obtain real values after the simulation within a margin of error, but obtaining this values by minimizing the computational cost. This has also been studied in this report

After the program development, a functionality of obtaining the temperature field at a desired point in time was implemented. With that functionality, a RESULTS.dat document was obtained in which the temperature field at that point in time was represented in a matrix, ready to get used by a MATLAB code originally developed that represents the temperature field as a color map. Using that, the temperature distributions from Figure A.6 were found.

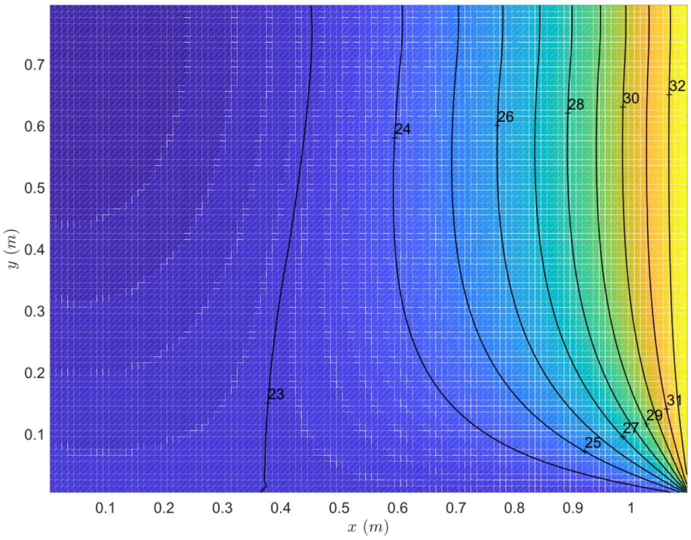


Figure A.6: Temperature distribution at $t = 5000$ s

A.6 Numerical properties' analysis

In order to evaluate the efficiency of the code as well as the relationship between the inputs - precision, mesh size, time step - and the execution time and the results. This evaluation would help to get the optimal inputs in order to run the code as fast as possible by getting realistic results.

A.6.1 Tolerance

In order to evaluate the efficiency of the code, the tolerance imposed to the code when the convergence is analysed is an important factor that determines the processing time of the code. In fact, it can also determine whether the code is able to converge or not, since the convergence criterion from the code may tend to a limit value (in this case, the convergence criterion is set to be the norm of the value of the discretization equation for each control point, what leads to a final value of $\approx 3 \cdot 10 - 9$) that will limit the maximum tolerance for a determined mesh and timestep.

In this case, the fixed properties were set with the following values:

- $N_x = 99 \rightarrow \Delta_x = 0.0111\ m$
- $N_y = 72 \Rightarrow \Delta_y = 0.0111\ m$
- $\Delta_t = 1.0\ s$

When the code was ran with the data from Table A.3, the results available in Figure A.7, Figure A.8 were obtained. What can be observed from the results is the fact that they are as they were expected to be, since for a smaller tolerance, the computing time gets bigger, while the change in the results can just be noticed for a 10^{-2} tolerance and smaller. For all the other cases, the results are exactly the same for the code developed. Hence, a tolerance smaller than 10^{-5} is **not necessary** in the conditions studied in this analysis.

Tolerance	10	1	10^{-1}	10^{-2}	10^{-3}
Computing time (s)	7.54576	15.7812	24.8119	33.4917	42.8075
$\mathbf{T(x_1, 5000\ s)}$ ($°C$)	23.6756	24.4631	24.5363	24.5425	24.543
$\mathbf{T(x_2, 5000\ s)}$ ($°C$)	24.6279	25.3932	25.4596	25.4647	25.4652
Tolerance	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
Computing time (s)	52.6539	62.9471	72.2749	82.4067	92.1626
$\mathbf{T(x_1, 5000\ s)}$ ($°C$)	24.5431	24.5431	24.5431	24.5431	24.5431
$\mathbf{T(x_2, 5000\ s)}$ ($°C$)	25.4652	25.4652	25.4652	24.5431	25.4652

Table A.3: Data from the convergence analysis

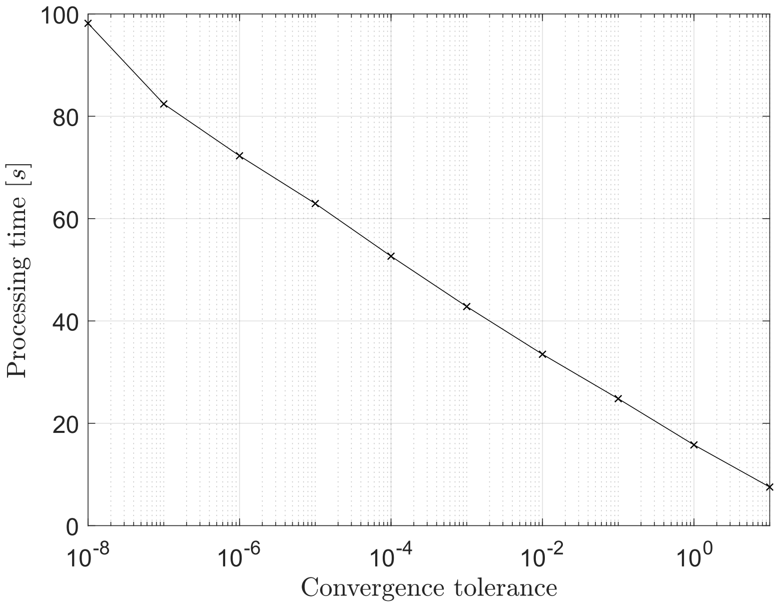


Figure A.7: Computing time from 0 to 5000 s as a function of the convergence tolerance, by keeping constant Δ_t, Δ_x and Δ_y.

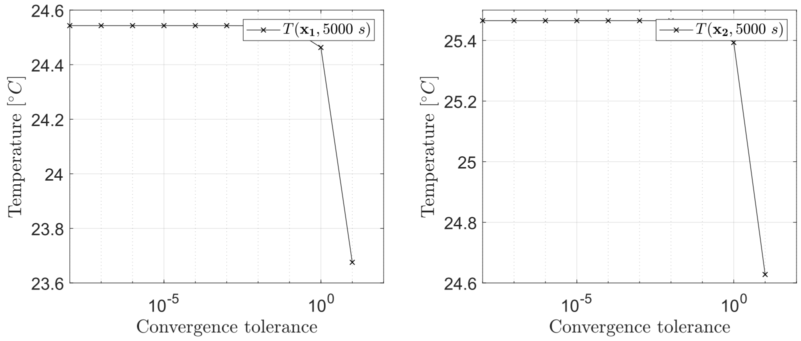


Figure A.8: Temperature in the required points (x_1 and x_2) at $t = 5000\ s$ as a function of the convergence tolerance, by keeping constant Δ_t, Δ_x and Δ_y.

A.6.2 Time step

In this case, the time step Δ_t was changed from rather small values (e.g $\Delta_t = 0.2\ s$) given the complexity of the problem to large values (e.g $\Delta_t = 10\ s$) just to demonstrate that for a larger time step, the precision of the results will be poorer, whereas the processing time will reduce a remarkable amount of time. These simulation data is available in Table A.4, whereas the graphical results can be seen in Figure A.9,Figure A.10.

By keeping $\Delta_x = 0.0111\ m$, $\Delta_y = 0.0111\ m$ and a tolerance of 10^{-5}, the results found are consistent with what it was expected. In fact, the variation in time step for the values of $0.1\ s,\ 0.2\ s$ is not notable, while in the computing time, the difference is around a minute for simulations with a final time of 5000 s and a quite small mesh (7128 control volumes), what would results in greater time delays in finer meshes and longer simulations. This is the reason why a time step of 1 s or 0.5 s would be the best options when the decision on what time step is optimal is taken.

Time step	0.1	0.2	0.5	1.0	5.0	10.0
Computing time (s)	188.929	122.813	78.8031	63.1045	46.8505	43.4436
$\mathbf{T(x_1, 5000\ s)}$ $(^\circ C)$	24.5433	25.5433	24.5432	24.5431	24.542	24.5406
$\mathbf{T(x_2, 5000\ s)}$ $(^\circ C)$	25.4655	25.4654	25.4653	25.4652	25.4641	25.4628

Table A.4: Data from the time step analysis

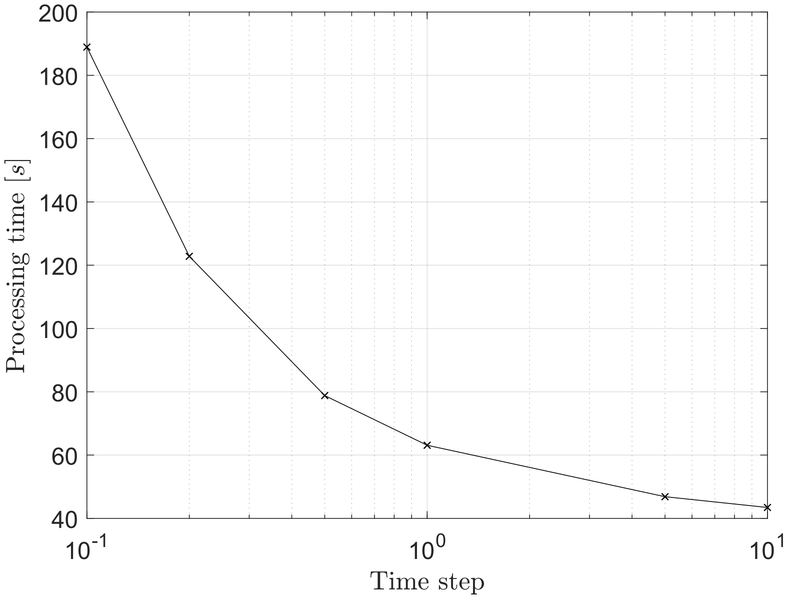


Figure A.9: Computing time from 0 to 5000 s as a function of the time step, by keeping constant the tolerance, Δ_x and Δ_y.

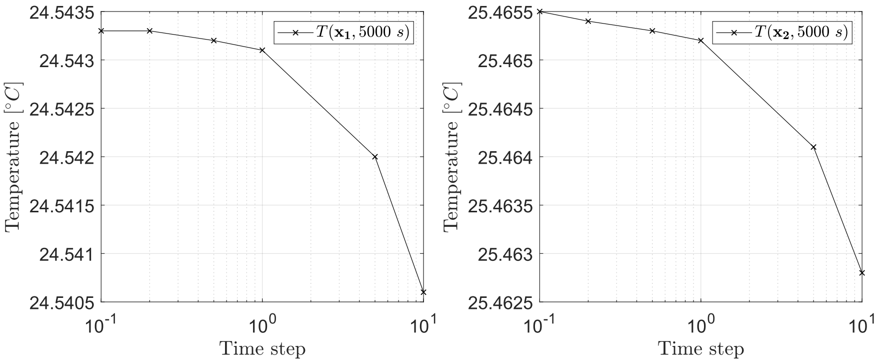


Figure A.10: Temperature in the required points (x_1 and x_2) at $t = 5000$ s as a function of the time step, by keeping constant the tolerance, Δ_x and Δ_y

A.6.3 Mesh size

One of the most important parameters for a numerical solution is the size of the discrete domain, which can be easily parametrized by the number of control volumes. In this case, the number of control volumes has varied from 88 to 35000, in order to have an important range of values in which evaluate the results, which are available in Table A.5.

Number of control volumes	88	352	792	1408	2200	3168	4312	5632	7128	8800	35000
Computing time (s)	0.308343	0.97062	2.60933	5.00929	8.08852	18.0348	23.6782	41.6088	62.4084	79.3887	1533.78
$\mathbf{T(x_1, 5000\ s)}$ $(°C)$	24.9627	24.4959	24.6513	24.4883	24.5866	24.4848	24.5587	24.4851	25.5431	24.4855	24.4854
$\mathbf{T(x_2, 5000\ s)}$ $(°C)$	24.9464	25.4941	25.2771	25.5067	25.3936	25.3185	25.4466	25.370	25.4652	25.4116	25.4211

Table A.5: Data from the mesh size analysis

This results can be summarized in Figure A.11, Figure A.12. In the first of the both graphical representations, as it could be expected, the strong relationship between number of control volumes and time is represented. In this case, the time does not correspond to the total time of the system, yet to 5000 s which corresponds to the middle point of the simulation requested. Even though there are some small variations on the results when introducing a big number of control volumes, the code seems to converge to a finite value for both points at 5000 s.

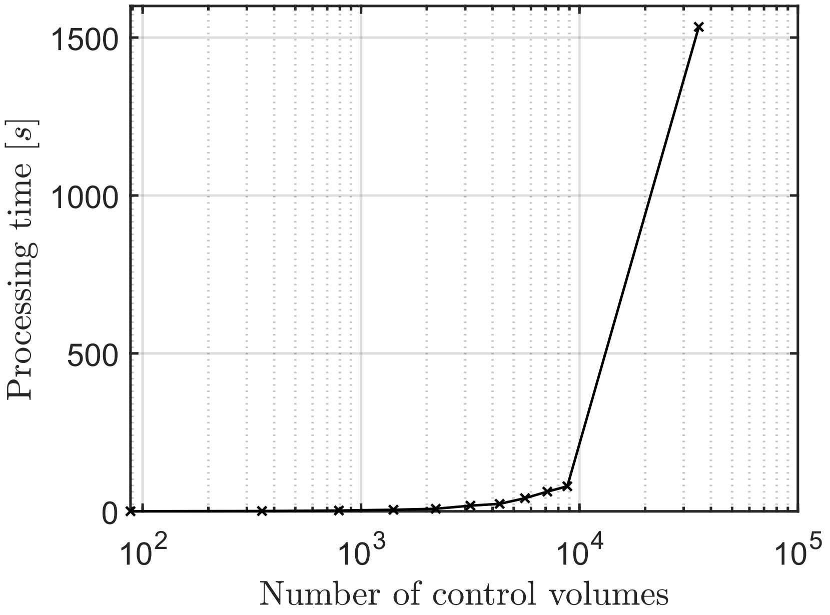


Figure A.11: Computing time from 0 to 5000 s as a function of the number of control volumes, by keeping constant the tolerance and Δ_t.

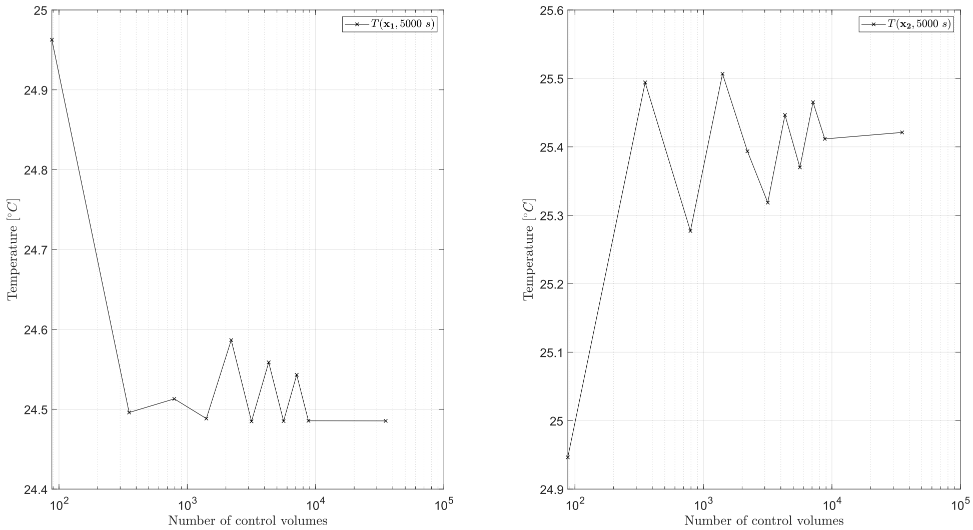


Figure A.12: Temperature in the required points (x_1 and x_2) at $t = 5000\ s$ as a function of the number of control volumes, by keeping constant the tolerance and Δ_t

A.7 Conclusion

This report summarizes the development and analysis of a **Finite Volume Method** (FVM) code that solves (A.2) in the case described in the Introduction. It is important to remark the fact that this problem has a lot of interest, since all the elementary boundary conditions for a problem of this kind are applied (constant and uniform temperature, variable yet uniform temperature, heat flux and a fluid moving), what makes the implementation more interesting and complete in order to fully understand the theoretical behaviour of all of them.

After studying the behaviour of the code in section A.6, once seen the results in all the figures in the corresponding section, probably a configuration of Table A.6 could be a good standard for this corresponding case.

Tolerance	$\leq 10^{-5}$
Time step	$\leq 1.0\ s$
Number of control volumes	> 8800

Table A.6: Minimum specifications for converged results

Once implemented the code, the analysis part was very important to fully understand the Numerical Methods, since it was my first contact with them. Understanding the basic concepts behind the results in Table A.3,Table A.4 and Table A.5 is important for any beginner in Numerical Methods, and this is one of the best methods to do it.

Appendix B

Smith-Hutton problem

B.1 Introduction

A solution for the Smith-Hutton problem has to be obtained, being (B.1) the PDE needed to solve.

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla\left(\rho\vec{u}\phi\right) = \nabla\left(\Gamma\nabla\phi\right) + S \tag{B.1}$$

B.1.1 Problem definition

In a rectangular domain with a velocity field (B.2), with the boundary conditions from Table B.1, solve the ϕ field knowing previous numerical results.

$$\begin{aligned} u(x,y) &= 2y(1-x^2) \\ v(x,y) &= -2x(1-y^2) \end{aligned} \tag{B.2}$$

Boundary condition	x-range	y-range
$\phi = 1 + \tanh\left[(2x+1)\alpha\right]$	$-1 < x < 0$	$y = 0$
$\phi = 1 - \tanh\left[\alpha\right]$	$x = -1$	$0 < y < 1$
	$-1 < x < 1$	$y = 1$
	$x = 1$	$0 < y < 1$
$\frac{\partial\phi}{\partial y}$	$0 < x < 1$	$y = 0$

Table B.1: Boundary conditions, with $\alpha = 10$

B.2 Discretization equation

Beginning from (B.1) and by considering a permanent problem without source terms, the process yields (B.3), which ends to the standard discretization equation format (B.4), where the coefficients can be calculated using (B.5).

$$\frac{\partial}{\partial x}\left(\Gamma\phi u\right) + \frac{\partial}{\partial y}\left(\Gamma\phi v\right) = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} \tag{B.3}$$

$$a_P\phi_P = a_E\phi_E + a_W\phi_W + a_N\phi_N + a_S\phi_S + b \tag{B.4}$$

$$a_E = D_e A(|\mathrm{Pe}_e|) + \max(-F_e, 0) \tag{B.5a}$$

$$a_W = D_w A(|\mathrm{Pe}_w|) + \max(F_w, 0) \tag{B.5b}$$

$$a_N = D_n A(|\mathrm{Pe}_n|) + \max(-F_n, 0) \tag{B.5c}$$

$$a_S = D_s A(|\mathrm{Pe}_s|) + \max(F_s, 0) \tag{B.5d}$$

$$a_P = a_E + a_W + a_N + a_S \tag{B.5e}$$

Where $A(|\mathrm{Pe}|) = \max(0, (1 - 0.1|\mathrm{Pe}|)^5)$

B.3 Algorithm

1. Data input
   - Physical properties ($\Gamma,\ \rho,\ L_x,\ L_y,\ u,\ v$)
   - Numerical properties ($N_x,\ N_y,\ \delta$)
2. Initial calculations ($\Delta_x,\ D_i,\ F_i,\ \mathrm{Pe}_i,\ \phi_{in}$)
3. Discretization coefficients
4. Start Line by Line
   (a) For each line, P and Q calculations
   (b) TDMA
   (c) It is the norm lower than the tolerance?
      - If yes, go to 5. If not, set $\phi_{ant} = \phi_{act}$, and return to 4.
5. Print results
6. End

B.4 Mesh analysis

As it was done previously for the other problems, a mesh convergence analysis must be developed in order to prove that the solution does not depend on the mesh size. Hence, as this is meant to be a code that is able to compute this phenomena for all P, three different Péclet Numbers (represented by ρ/Γ) were used, providing a wide range of convergence analysis. These results are summarized in Figure B.1. Furhtermore, an analysis on execution time and number of iterations can be also found in Figure B.2

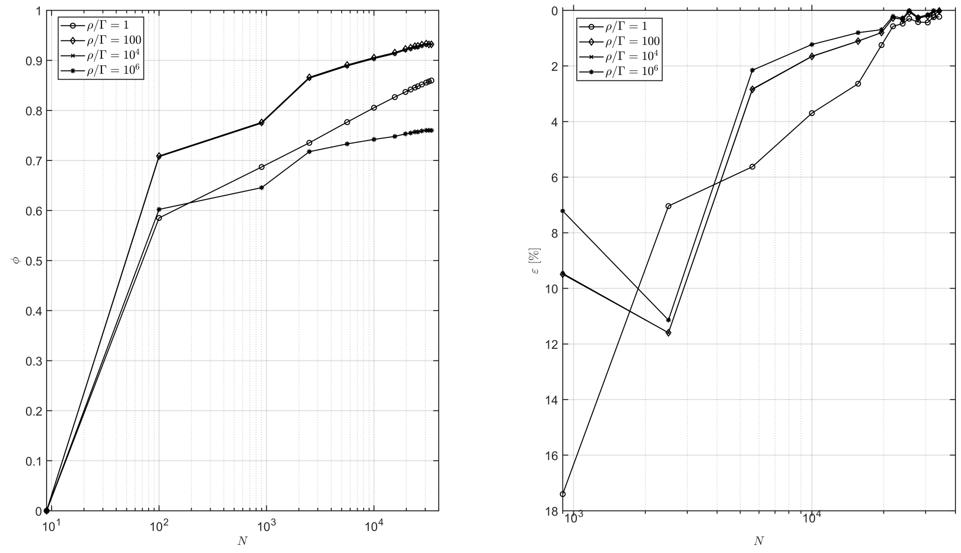


Figure B.1: Convergence analysis for Problem 4 at different ρ/Γ numbers

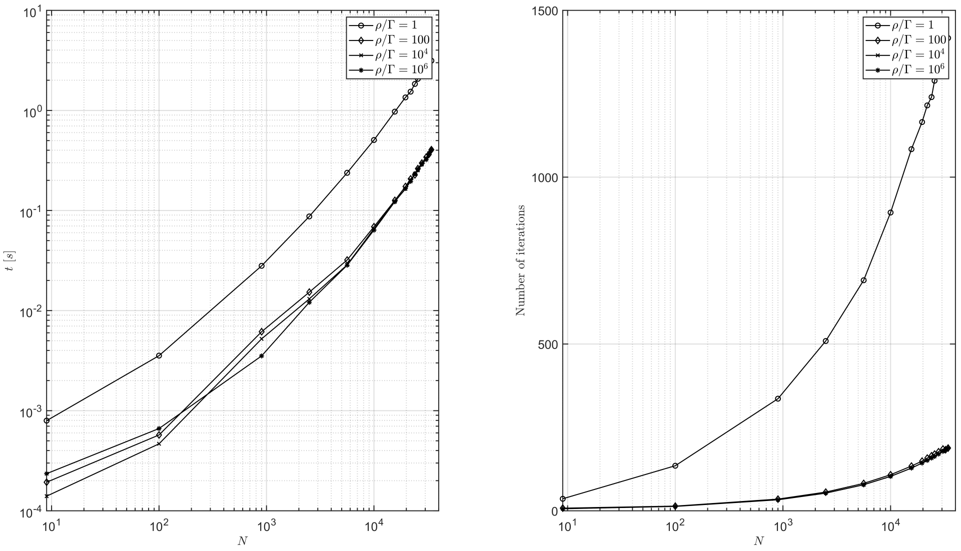


Figure B.2: Execution time and number of iterations analysis for Problem 4 at different ρ/Γ numbers

The results from this analysis will be found in the following tables [1].

[1] All the results in Problem 4 are done using a 10^{-8} convergence tolerance.

N	$\phi(x=0.5,y=0)$	ε [%]	t [s]	Number of iterations
9	$2.06 \cdot 10^{-9}$	-	0.000795	36
100	0.5851	$2.83 \cdot 10^{10}$	0.003547	135
900	0.686907	17.39993	0.027999	336
2500	0.735266	7.04011	0.087269	509
5625	0.776631	5.62586	0.238093	691
10000	0.80535	3.69790	0.506782	894
15625	0.826612	2.64009	0.972182	1084
19600	0.83693	1.24859	1.34992	1165
21904	0.841737	0.57400	1.53892	1215
24025	0.845745	0.47616	1.83908	1240
25600	0.848266	0.29808	2.06796	1289
27889	0.851843	0.42168	2.26241	1313
30625	0.855558	0.43611	2.71766	1359
32400	0.85767	0.24686	2.8242	1407
34225	0.859672	0.23342	3.13948	1417

Table B.2: Convergence analysis for $\rho/\Gamma = 1$

N	$\phi(x=0.5,y=0)$	ε [%]	t [s]	Number of iterations
9	$2.06 \cdot 10^{-9}$	-	$1.93 \cdot 10^{-4}$	8
100	0.708852	34391092251	$5.72 \cdot 10^{-4}$	14
900	0.776138	9.49224944	0.006156	35
2500	0.866145	11.59677789	0.015299	56
5625	0.890771	2.84317291	0.032047	82
10000	0.905539	1.657889626	0.068858	108
15625	0.915578	1.108621495	0.126216	134
19600	0.922903	0.800041067	0.173759	149
21904	0.925482	0.279444319	0.20696	158
24025	0.928424	0.317888408	0.226862	165
25600	0.928976	0.059455594	0.262069	170
27889	0.931583	0.280631577	0.297801	177
30625	0.933507	0.206530175	0.340923	185
32400	0.931879	0.174396121	0.363627	183
34225	0.932066	0.020066983	0.404876	188

Table B.3: Convergence analysis for $\rho/\Gamma = 100$

N	$\phi(x=0.5, y=0)$	ε [%]	t [s]	Number of iterations
9	$2.06 \cdot 10^{-9}$	-	$1.40 \cdot 10^{-4}$	6
100	0.707589	34329815779	0.000467	13
900	0.774558	9.464392465	0.00522	33
2500	0.864394	11.59835674	0.013089	53
5625	0.888869	2.83146343	0.028643	78
10000	0.903565	1.653336993	0.065814	103
15625	0.913541	1.104071096	0.122917	128
19600	0.920847	0.799745167	0.164457	143
21904	0.923414	0.278765093	0.201745	151
24025	0.926346	0.317517387	0.22786	158
25600	0.92689	0.058725357	0.251848	163
27889	0.929489	0.280400047	0.28829	170
30625	0.931403	0.205919597	0.333183	178
32400	0.931879	0.051105698	0.372556	183
34225	0.932066	0.020066983	0.407184	188

Table B.4: Convergence analysis for $\rho/\Gamma = 10000$

N	$\phi(x=0.5, y=0)$	ε [%]	t [s]	Number of iterations
9	$2.06 \cdot 10^{-9}$	-	$2.35 \cdot 10^{-4}$	6
100	0.602228	29218057780	0.000663	13
900	0.645687	7.21636988	0.003526	33
2500	0.717606	11.1383689	0.012093	53
5625	0.733063	2.153967497	0.028404	78
10000	0.742075	1.229362279	0.063702	103
15625	0.748042	0.804096621	0.12247	128
19600	0.753234	0.694078675	0.165739	143
21904	0.754866	0.216665737	0.195337	151
24025	0.756994	0.281904338	0.233567	158
25600	0.757066	0.009511304	0.258393	163
27889	0.758947	0.248459183	0.293912	170
30625	0.760132	0.156137385	0.322531	178
32400	0.760203	0.009340483	0.363381	183
34225	0.76003	0.022757079	0.403192	188

Table B.5: Convergence analysis for $\rho/\Gamma = 1000000$

B.5 Results

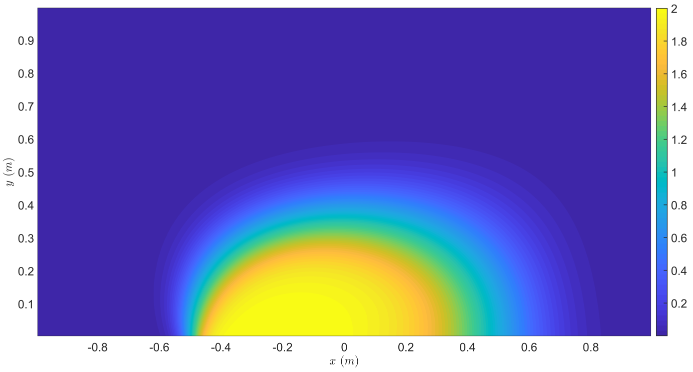


Figure B.3: ϕ field for a 180 × 180 mesh with $\rho/\Gamma = 1$

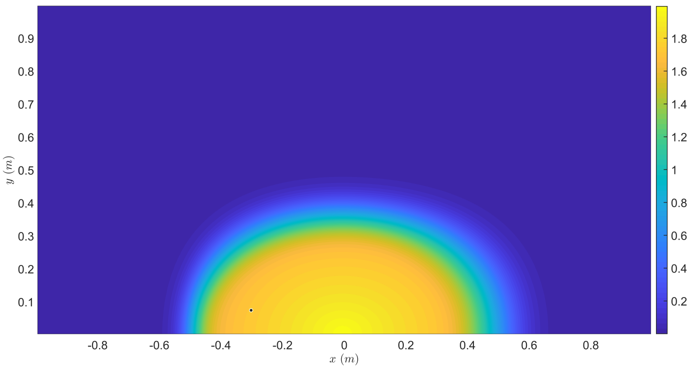


Figure B.4: ϕ field for a 180 × 180 mesh with $\rho/\Gamma = 10^6$

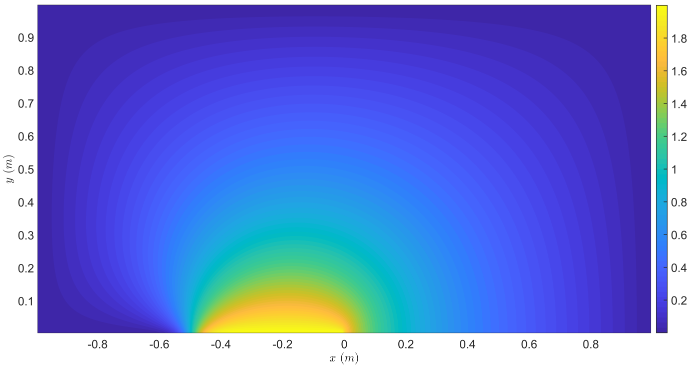


Figure B.5: ϕ field for a 180 × 180 mesh with $\rho/\Gamma = 10^{-3}$

Appendix C

Lid-Driven Cavity. Comparison between FSM and SIMPLER

C.1 Introduction

The purpose of this chapter is to compare both **Fractional Step Method** (FSM, henceforth) [4] and **Semi-Implicit Method for Pressure-Linked Equations Revised** (SIMPLER, henceforth) [3] in its behaviour in the numerical resolution of the Navier-Stokes[1] equations (C.1) using the Finite Volume Method. In order to do so, Ghia's publication on the numerical solution of the problem known as **Driven Cavity** [6] will be used to check the correction of the results obtained in this report.

$$\begin{cases} \frac{\partial \rho}{\partial t} + \rho \frac{\partial u_i}{\partial x_i} = 0 \\ \rho \frac{\partial u_i}{\partial t} + \rho u_j \frac{\partial u_i}{\partial x_j} = \mu \frac{\partial^2 u_i}{\partial x_j \partial x_j} - \frac{\partial p}{\partial x_i} \end{cases} \tag{C.1}$$

The problem can be described as a 2D problem based in which a square domain of analysis in which all the boundaries are solid and isolating walls, where the bottom, left and right are static, whereas the top wall is moving at a horizontal velocity $u = 1$. This sets the boundary condition outline available in Figure C.1. Regarding the pressure, a non-gradient of the pressure in the normal direction to the wall also has to be included.

This problem is solved for $\text{Re} = 100, 400, 1000, 3200, 5000, 7500, 10000$, which will give a wide view on the response with both resolution schemes for the Navier-Stokes equations, where Re can be defined by (C.2).

$$\text{Re} = \frac{\rho V_0 L}{\mu} \tag{C.2}$$

In order to apply these properties defined by the Reynolds number, all will be set to 1, with the exception of ρ, which will be considered to be equal to Re

C.2 SIMPLER Resolution

The SIMPLER method (described by Patankar, in 1979) stands for the revision of the first described by Patankar and Spalding in 1972 Semi-Implicit Method for Pressure-Linked Equa-

[1] Remark that the energy equation, which is part of the Navier-Stokes equations , will not be used in this report. Nevertheless, its implementation will not be difficult when the flow field is calculated

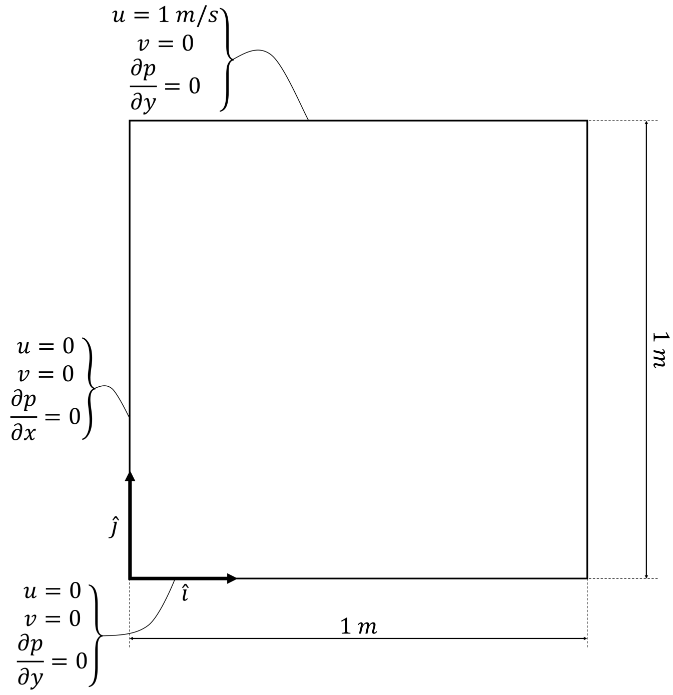

$u = 1\,m/s$
$v = 0$
$\frac{\partial p}{\partial y} = 0$
$u = 0$
$v = 0$
$\frac{\partial p}{\partial x} = 0$
$\hat{j}$
$\hat{\imath}$
$u = 0$
$v = 0$
$\frac{\partial p}{\partial y} = 0$
$1\,m$
$1\,m$

Figure C.1: Boundary conditions outline for Lid-Driven Cavity[6].

tions (SIMPLE), in order to avoid some difficulties in the resolution of the problem, which might be solved by using underrelaxation (See [3]) and to increase the rate of convergence.

C.2.1 Staggered grid

In order to avoid the checkerboard problem (see [3] for further information), either a staggered or a collocated mesh has to be used. In this case, the staggered grid will be used, which is based in the displacement of the flow field control volumes either in the horizontal direction for the u velocity field or in the vertical direction for the v velocity field. This kind of grid, in which some bounded nodes for all three grids which are used on this code are used in order to simplify the imposition of boundary conditions.

In this corresponding case, a non uniform grid will be used in order to give more importance to the points next to the boundary, in which a boundary layer will be generated, which study and more detailed simulation is of great importance for the numerical solver.

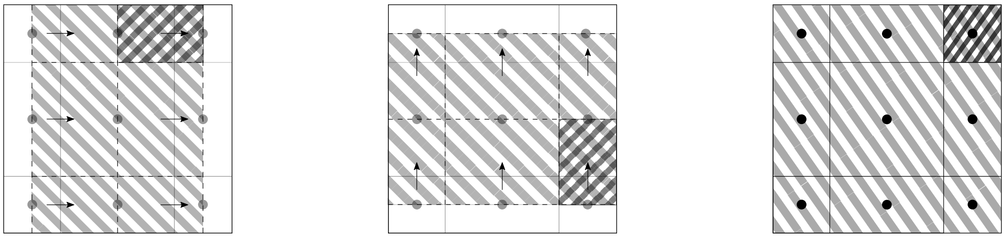
In order to do so, a full cosine distribution was used (C.3) in both x and y directions, which is represented in Figure C.2, with its corresponding staggered grid situation.

$$x_i = \frac{L}{2}\left[1 - \cos\left(\frac{i}{N_x}\pi\right)\right] \tag{C.3a}$$

$$y_i = \frac{L}{2}\left[1 - \cos\left(\frac{i}{N_y}\pi\right)\right] \tag{C.3b}$$

C.2.2 Algorithm

1. Data introduction



(a) $x-$staggered grid. (b) $y-$staggered grid. (c) Main grid.

Figure C.2: Representation of the staggered grid. In all representations, the dark cell corresponds to just a control volume, whereas the light cells correspond to all the domain.

   (a) Physical properties

   (b) Numerical properties

2. Previous calculations

3. Set the guessed velocity field: $u_g(x, y) = u_0(x, y)$, $v_g(x, y) = v_0(x, y)$. This field will be considered to be zero in all points in which its value is unknown.

4. Calculation of the momentum coefficients of a convection-diffusion equation (C.4), where $\Gamma = \mu$, $S = -\nabla p$ and $\phi = u$ or $\phi = v$, by considering a staggered grid, by taking in consideration the boundary conditions from Figure C.1, using any of the schemes described in [28] and previously used in [29].

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla\left(\rho\vec{u}\phi\right) = \nabla\left(\Gamma\nabla\phi\right) + S \tag{C.4}$$

   The evaluation of the velocities in the interfaces of the non-staggered control volumes is done using a **Central Difference Scheme** (henceforth, CDS) in both $u(x, y)$ and $v(x, y)$.

5. Calculation of the pseudovelocities $\hat{u}$, $\hat{v}$ using (C.5) with the guessed velocity field, where P_u stands for the node of the staggered grid equivalent to the conventional grid P.

$$\hat{u}_{P_u} = \frac{\sum a_{nb}u_{nb} + b}{a_{P_u}} \tag{C.5}$$

6. Solve the pressure field p^*, by computing the coefficients as (C.6)

$$a_E = \rho_e d_e \Delta_y \tag{C.6a}$$

$$a_W = \rho_w d_w \Delta_y \tag{C.6b}$$

$$a_N = \rho_n d_n \Delta_x \tag{C.6c}$$

$$a_S = \rho_s d_s \Delta_x \tag{C.6d}$$

$$a_P = a_E + a_W + a_N + a_S \tag{C.6e}$$

$$b = \frac{(\rho_P^0 - \rho_P)\Delta_x\Delta_y}{\Delta_t} + \left[(\rho u^*)_w - (\rho u^*)_e\right]\Delta_y + \left[(\rho v^*)_s - (\rho v^*)_n\right]\Delta_y \tag{C.6f}$$

7. Solve the momentum equations to obtain u^*, v^*, by considering the pressure gradient term as ∇p^*

8. Solve de pressure field p', by computing the coefficients as (C.6), replacing the u^* and v^* by its starred versions.

9. Correct the u^* and v^* fields with the pressure gradient using (C.7).

$$u_{P_u} = u^*_{P_u} + d_e(p'_P - p'_E) \tag{C.7}$$

10. Has the velocity field converged? In this case, a relative variation of both velocity fields (C.8) was calculated and it was expected to be under a certain tolerance, which corresponds to a numerical input of the solver.

$$\max\left(\frac{|u(x,y) - u_{li}(x,y)|}{|u_{li}|}\right) < \delta_u \tag{C.8}$$

Where u_{li} stands for the last iteration velocity. In the literature, the notation used is u^*. Nevertheless, this might generate some confusion in this context, therefore, this new notation is used.

   - Yes: Step 12
   - No: Back to Step 4 with the seed velocities as the guessed ($u_g(x,y) = u(x,y)$, $v_g(x,y) = v(x,y)$)

11. If there is convergence on all the fields, considering the research of a stationary field, next time step?

   - Yes: Back to point 1 with the last temperature. This will be accomplished when (C.8) is achieved in the time perspective.
   - No: Step 13

12. Last calculations and print results

13. End

C.2.3 Results

In order to validate the results generated by this solver, two different solvers for linear systems of equations were considered:

- Gauss-Seidel Line-by-Line (henceforth, LBL), in all cases.
- Conjugate Gradient Method (henceforth, CG) for solving pressure fields, and LBL for velocity fields.

The use of the latter is justified by the fact that the number of iterations required by the first one to converge was so large that the simulating time was too large, as observed in [30], which completely justifies this implementation. Nevertheless, at the date in which this report has been written the CG algorithm implementation is in revision, which is the reason why the results are found using the LBL algorith.

Re = 100

Using the Gauss-Seidel line-by-line algorithm, for Re = 100, the results from Figure C.3 were obtained. In the figure, this results are compared with its corresponding points in [6].

On the other hand, similar analysis the same analysis can be done for CG algorithm.

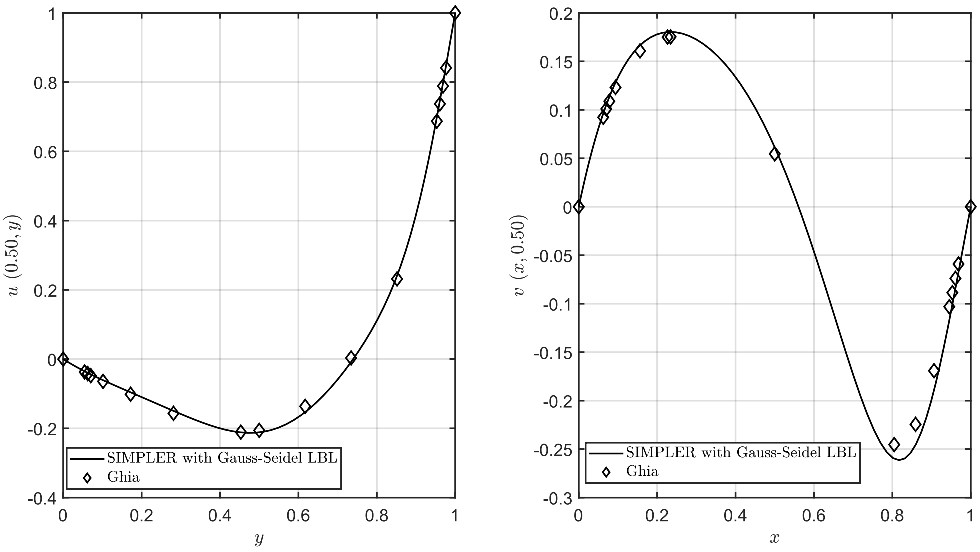
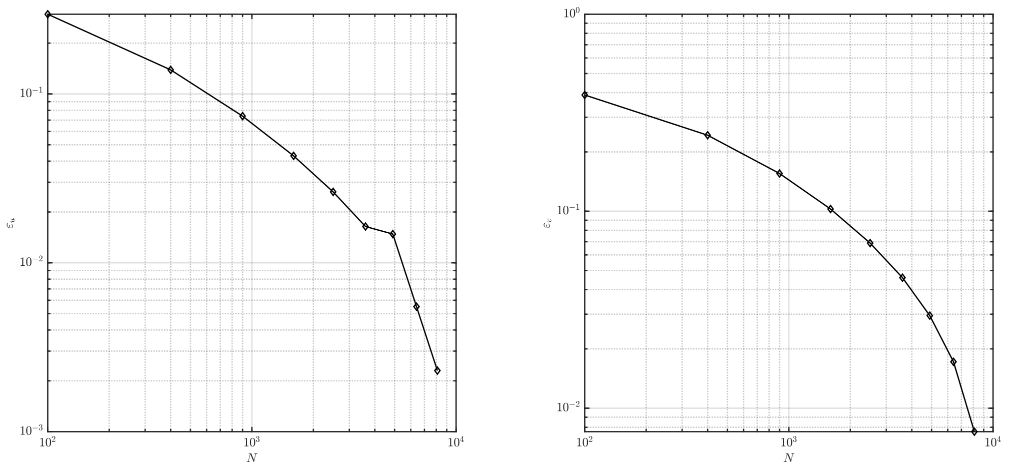


Figure C.3: Comparison between the results at the specified points calculated using the SIMPLER algorithm with the pressure solved using a LBL method and the results given in [6] using a 80 × 80 mesh.

Convergence analysis

In order to evaluate the results, a convergence analysis was done for both mesh size and timestep, since convergence has to be ensured for both discretized domains. From the point of view of the mesh, the results are available at Table C.1, and its graphical representation can be found in Figure C.4.



(a) ε_u with respect to $N = 10000$.

(b) ε_v with respect to $N = 10000$.

Figure C.4: Mesh size analysis with Re = 100.

Computational cost analysis

Since relaxation is on concern, its affect to the computational costs was studied using different mesh sizes for a constant timestep ($\Delta_t = 0.01\ s$), which results are available at Table C.3

and are represented in C.5.

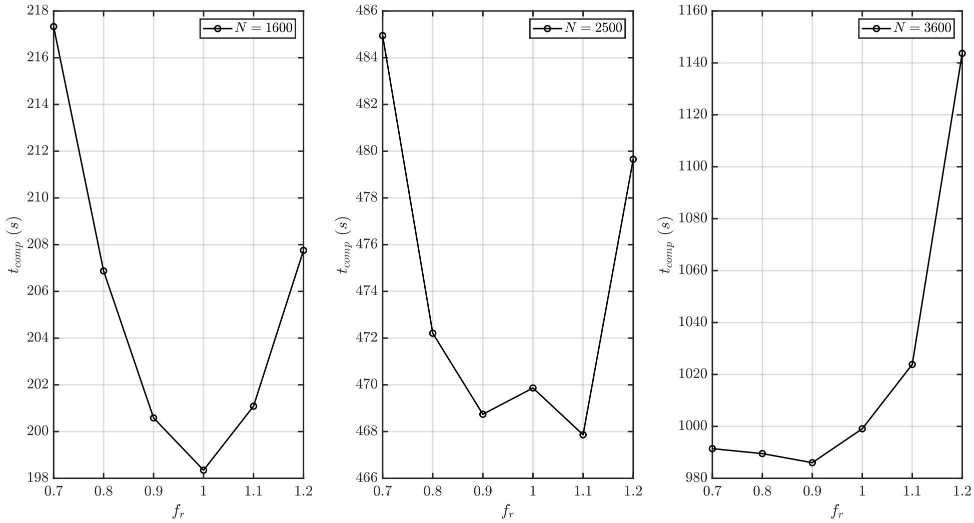


Figure C.5: Influence of the relaxation factor in different number of control volumes ($N = 1600, 2500, 3600$) per a timestep $\Delta_t = 0.01\ s$.

On the other hand, computational time per timestep has been calculated for this problem (Figure C.6). In order to see the exact results, visit Table C.4.

Re = 1000

For this Reynolds number, the development was done for both LBL (Figure C.7) and CG (Figure C.8), which are almost identical, being this an indication on the fact that the solution using both solvers is equivalent.

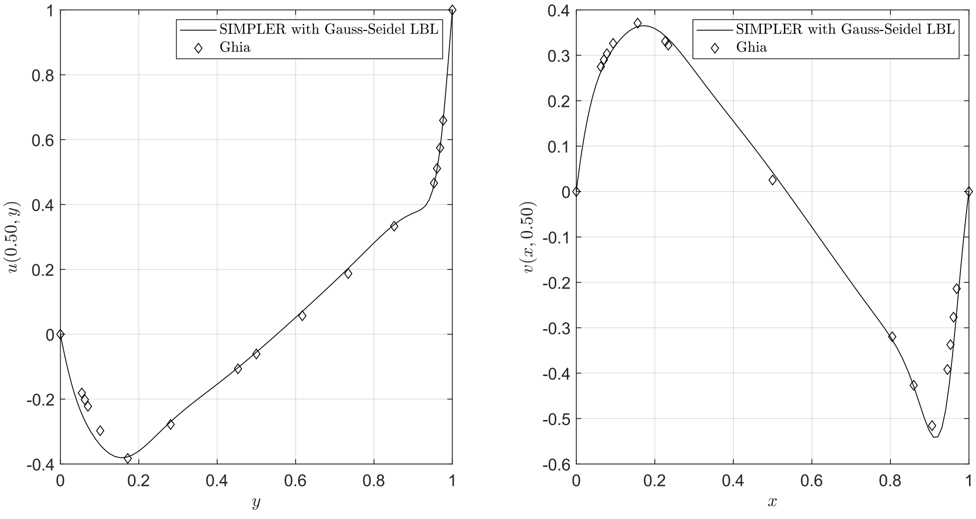


Figure C.7: Comparison between the results at the specifiec points calculated using the SIMPLER algorithm using the LBL method and the results given in [6] using a 80×80 mesh.

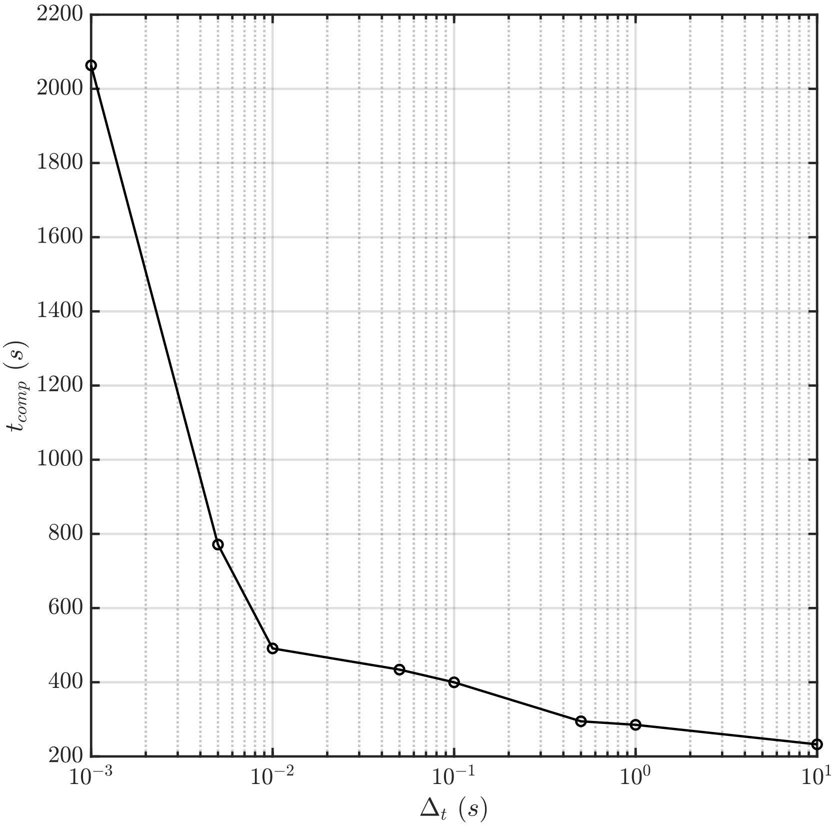


Figure C.6: Influence of the timestep Δ_t for $N = 2500$ and $f_r = 0.90$.

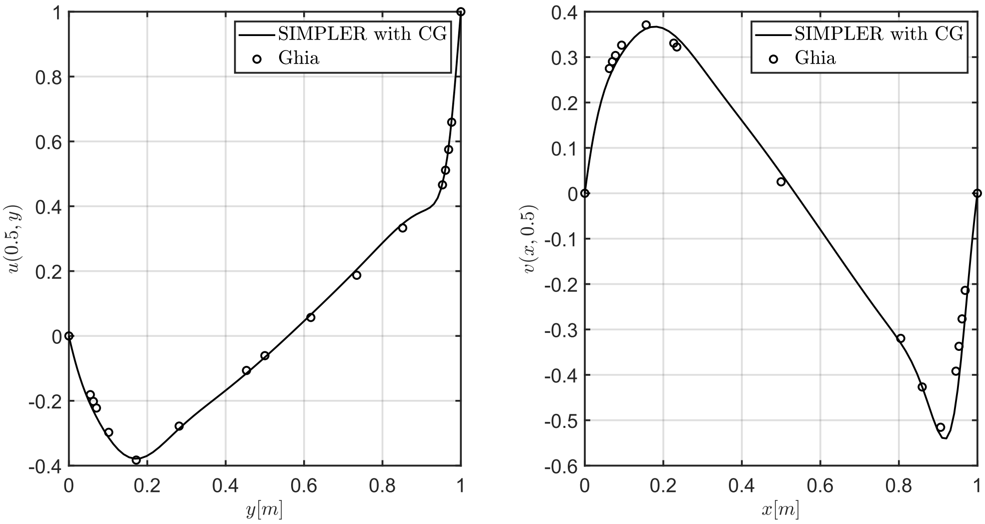


Figure C.8: Comparison between the results at the specifiec points calculated using the SIMPLER algorithm using the CG algorithm and the results given in [6] using a 80 × 80 mesh.

C.3 FSM Resolution

C.3.1 Method derivation

Beginning from the Navier-Stokes equations (C.1) in its incompressible form (C.9),

$$\nabla \cdot \vec{u} = 0 \tag{C.9a}$$

$$\rho\frac{\partial \vec{u}}{\partial t} + \rho(\vec{u}\cdot\nabla)\vec{u} = -\nabla p + \mu\nabla^2\vec{u} \tag{C.9b}$$

(C.9b) can be rewritten as (C.10), in which there is a term $\vec{R}(\vec{u})$ that includes momentum transport and momentum diffusion.

$$\rho\frac{\partial \vec{u}}{\partial t} = \vec{R}(\vec{u}) - \nabla p \tag{C.10}$$

Where:

$$\vec{R}(\vec{u}) = -\rho(\vec{u}\cdot\nabla)\vec{u} + \mu\nabla^2\vec{u}$$

Hence, if (C.10) is discretized in the time domain using an Adams-Bashforth scheme for $\vec{R}(\vec{u})$, the new set of equations becomes (C.11).

$$\nabla \cdot \vec{u}^{\ n+1} = 0 \tag{C.11a}$$

$$\rho\frac{\vec{u}^{\ n+1} - \vec{u}^{\ n}}{\Delta_t} = \frac{3}{2}\vec{R}(\vec{u}^{\ n}) - \frac{1}{2}\vec{R}(\vec{u}^{\ n-1}) - \nabla p^{\ n+1} \tag{C.11b}$$

Now, by applying the Helmholtz-Hodge Theorem (henceforth, HH) (C.12),

$$\vec{u}^{\ p} = \vec{u}^{\ n+1} + \frac{\Delta_t}{\rho}\nabla p^{n+1} \tag{C.12}$$

the momentum equation is projected onto this new HH space, which yields (C.13), an equation from which the predictor velocity can be extracted[2]

$$\frac{\rho}{\Delta_t}\left(\vec{u}^{\ p} - \frac{\Delta_t}{\rho}\nabla p^{n+1}\right) = \frac{3}{2}\vec{R}(\vec{u}^{\ n}) - \frac{1}{2}\vec{R}(\vec{u}^{\ n-1}) - \nabla p^{\ n+1}$$

$$\rho\frac{\vec{u}^{\ p} - \vec{u}^{\ n}}{\Delta_t} = \frac{3}{2}\vec{R}(\vec{u}^{\ n}) - \frac{1}{2}\vec{R}(\vec{u}^{\ n-1}) \tag{C.13}$$

Regarding the continuity equation (C.11a), it is also modified by the HH Theorem, yielding the problem's equation to solve (C.14).

$$\nabla \cdot \vec{u}^{\ n+1} \to \nabla \cdot \left(\vec{u}^{\ p} - \frac{\Delta_t}{\rho}\nabla p^{n+1}\right)$$

$$\nabla\vec{u}^{\ p} = \frac{\Delta_t}{\rho}\nabla^2 p^{n+1} \tag{C.14}$$

Hence, the projected velocity can be computed from (C.13) for all the field. Then, introducing its value onto (C.14) and solving it, the pressure field will be computed, which will yield, eventually, the flow field using (C.12).

[2] Recall that all the other variables have been computed at this point of the derivation

Evaluation of $\vec{R}(\vec{u})$

Consider the derivation in the x direction. Then, $R(u) = -(\rho\vec{u}\cdot\nabla)u + \mu\nabla^2 u$. Integrating this expression in the domain $\overline{\Omega}_x \coloneqq [\Delta_x, \Delta_y] \subset \Omega_x$,

$$\int_{\overline{\Omega}_x} R(u)\ d\overline{\Omega}_x = -\int_{\overline{\Omega}_x} (\rho\vec{u}\cdot\nabla)u\ d\overline{\Omega}_x + \int_{\overline{\Omega}_x} \mu\nabla^2 u\ d\overline{\Omega}_x$$

Now, by applying Gauss' Divergence Theorem (C.15), the integral form will be calculated (C.16).

$$\int_S f(x,y)\ dS = \int_V \nabla f(x,y)\ dV \tag{C.15}$$

$$\int_{\overline{\Omega}_x} R(u)\ d\overline{\Omega}_x = -\int_{\partial\overline{\Omega}_x} \rho\vec{u}\cdot\hat{n}u\ dA + \int_{\overline{\Omega}_x} \mu\nabla u\hat{n}\ dA \tag{C.16}$$

Integrating over each subdomain $\overline{\Omega}_x$, the value for each $R(u)$ will be found (C.17)

$$R(u_{P_u})\overline{\Omega}_x = -\left[\rho u_e^2\Delta_y - \rho u_w^2\Delta_y + \rho u_n v_n\Delta_x - \rho u_s v_s\Delta_x\right] +$$
$$+\mu_e\frac{u_{E_u}-u_{P_u}}{\Delta_x^{PE}}\Delta_y - \mu_w\frac{u_{P_u}-u_{W_u}}{\Delta_x^{PW}}\Delta_y + \mu_n\frac{u_{N_u}-u_{P_u}}{\Delta_y^{PN}}\Delta_x - \mu_s\frac{u_{P_u}-u_{S_u}}{\Delta_y^{PS}}\Delta_x$$

$$R(u_{P_u}) = \frac{1}{\overline{\Omega}_x}\left[\rho\left(\Delta_y(u_w^2-u_e^2)+\Delta_x(u_s v_s - u_n v_n)\right) + \frac{\mu_e\Delta_y}{\Delta_x^{PE}}u_{E_u} + \frac{\mu_w\Delta_y}{\Delta_x^{PW}}u_{W_u} + \right.$$
$$\left. + \frac{\mu_n\Delta_x}{\Delta_y^{PN}}u_{N_u} + \frac{\mu_s\Delta_x}{\Delta_y^{PS}}u_{S_u} - \left(\Delta_y\left(\frac{\mu_e}{\Delta_x^{PE}}+\frac{\mu_w}{\Delta_x^{PW}}\right)+\Delta_x\left(\frac{\mu_n}{\Delta_y^{PN}}+\frac{\mu_s}{\Delta_y^{PS}}\right)\right)u_{P_u}\right] \tag{C.17}$$

Now, the issue is the calculation of the lateral velocities from the staggered grid. Even though all velocities can be calculated using a CDS, this will not happen only for the vertical speeds, which will be computed using a weighted scheme. Then, (C.18) details how to compute all the different velocities.

$$u_e = \frac{u_{E_u}+u_{P_u}}{2} \tag{C.18a}$$
$$u_w = \frac{u_{W_u}+u_{P_u}}{2} \tag{C.18b}$$
$$u_n = \frac{u_{N_u}+u_{P_u}}{2} \tag{C.18c}$$
$$u_s = \frac{u_{S_u}+u_{P_u}}{2} \tag{C.18d}$$
$$v_n = \frac{v_{P_v}d_{wP}+v_{E_v}d_{Pe}}{\Delta_x} \tag{C.18e}$$
$$v_s = \frac{v_{S_v}d_{wP}+v_{SE_v}d_{Pe}}{\Delta_x} \tag{C.18f}$$

Regarding $R(v)$, following an equivalent derivation; it can be calculated using (C.19) .

$$R(v_{P_v}) = \frac{1}{\overline{\Omega}_y}\left[\rho\left(\Delta_y(u_w v_w - u_e v_e^2)+\Delta_x(v_s^2 - v_n^2)\right) + \frac{\mu_e\Delta_y}{\Delta_x^{PE}}v_{E_v} + \frac{\mu_w\Delta_y}{\Delta_x^{PW}}v_{W_v} + \right.$$
$$\left. + \frac{\mu_n\Delta_x}{\Delta_y^{PN}}v_{N_v} + \frac{\mu_s\Delta_x}{\Delta_y^{PS}}v_{S_v} - \left(\Delta_y\left(\frac{\mu_e}{\Delta_x^{PE}}+\frac{\mu_w}{\Delta_x^{PW}}\right)+\Delta_x\left(\frac{\mu_n}{\Delta_y^{PN}}+\frac{\mu_s}{\Delta_y^{PS}}\right)\right)v_{P_v}\right] \tag{C.19}$$

Resolution of the laplacian equation

Integrating (C.14) over its domain $\overline{\Omega} := [\Delta_x, \Delta_y] \subset \Omega$, corresponding to the control volume domain.

$$\int_{\overline{\Omega}} \nabla^2 p^{n+1} \, d\overline{\Omega} = \frac{\rho}{\Delta_t} \int_{\overline{\Omega}} \nabla \cdot \vec{u} \, d\overline{\Omega}$$

By applying the divergence theorem (C.15), the integral form (C.20) is found.

$$\int_{\partial\overline{\Omega}} \nabla p^{n+1} \hat{n} \, dS = \frac{\rho}{\Delta_t} \int_{\partial\overline{\Omega}} \vec{u}^p \cdot \hat{n} \, dS \tag{C.20}$$

Hence, the discretization equation (C.21) is found, with its corresponding coefficients (C.22).

$$a_P p_P^{n+1} = a_E p_E^{n+1} + a_W p_W^{n+1} + a_N p_N^{n+1} + a_S p_S^{n+1} + b_P \tag{C.21}$$

$$a_E = \frac{\Delta_y}{\Delta_x^{PE}} \tag{C.22a}$$

$$a_W = \frac{\Delta_y}{\Delta_x^{PW}} \tag{C.22b}$$

$$a_N = \frac{\Delta_x}{\Delta_y^{PN}} \tag{C.22c}$$

$$a_S = \frac{\Delta_x}{\Delta_y^{PS}} \tag{C.22d}$$

$$a_P = a_E + a_W + a_N + a_S \tag{C.22e}$$

$$b_P = -\frac{\rho}{\Delta_t} \left[(u_{P_u}^p - u_{W_u}^p)\Delta_y + (v_{P_v}^p - v_{S_v}^p)\Delta_x \right] \tag{C.22f}$$

Then, this equation has to be solved using Conjugate Gradient so as to obtain quick and accurate results.

C.3.2 Courant-Friedrichs-Levy condition

Even though the CFL condition is only strongly required in explicit numerical schemes in order to force them onto a valid and physically correct solution, it was also considered in this case, in order to make sure the proper values are computed. In order to do so, and as [4] provides, the maximum Courant number that can be provided to the numerical scheme is 0.35. In the case of the SIMPLER Algorithm, due to the fact that it is not fully explicit, the timesteps can be a bit larger, which are one of the key factors in choosing SIMPLER-like algorithms in the resolution in the RANS model of turbulence, which seeks the average value of the flow field. Nevertheless, and since the most common turbulence models (Large Eddy Simulator (LES) and Direct Numerical Solution (DNS)) are solved using explicit solvers, such as FSM, this method is implemented. In this case, the timestep will be defined using (C.23).

$$\Delta_t = \min \left[\min \left(0.35 \frac{\Delta_x}{|\mathbf{u}|} \right), \min \left(0.20 \frac{\rho \Delta_x^2}{\mu} \right) \right] \tag{C.23}$$

C.3.3 Results

In the case of FSM only the resolution using CG was considered. In this case, the timestep analysis was not done since the timesteps were determined using the CFL condition.

Re = 100

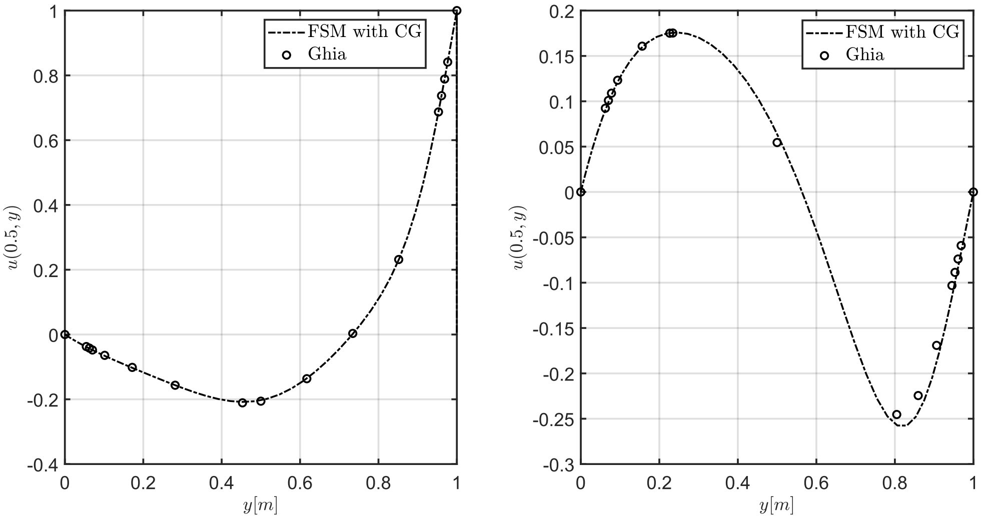


Figure C.9: Comparison between the results at the specific points calculated using the FSM algorithm using the CG algorithm and the results given by [6] using a 50 × 50 mesh in Re = 100.

Re = 400

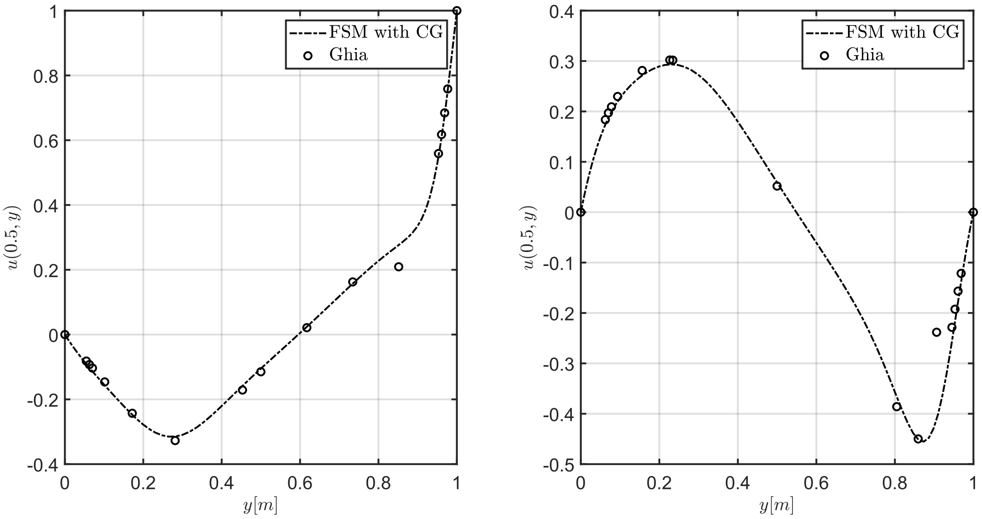


Figure C.10: Comparison between the results at the specific points calculated using the FSM algorithm using the CG algorithm and the results given by [6] using a 80 × 80 mesh in Re = 400.

Re = 1000

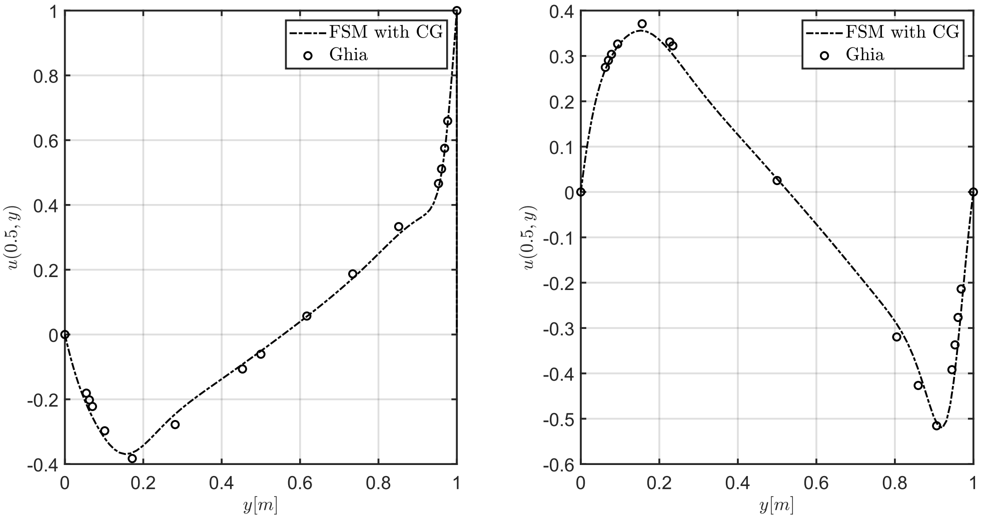


Figure C.11: Comparison between the results at the specific points calculated using the FSM algorithm using the CG algorithm and the results given by [6] using a 80 × 80 mesh in Re = 1000.

Re = 3200

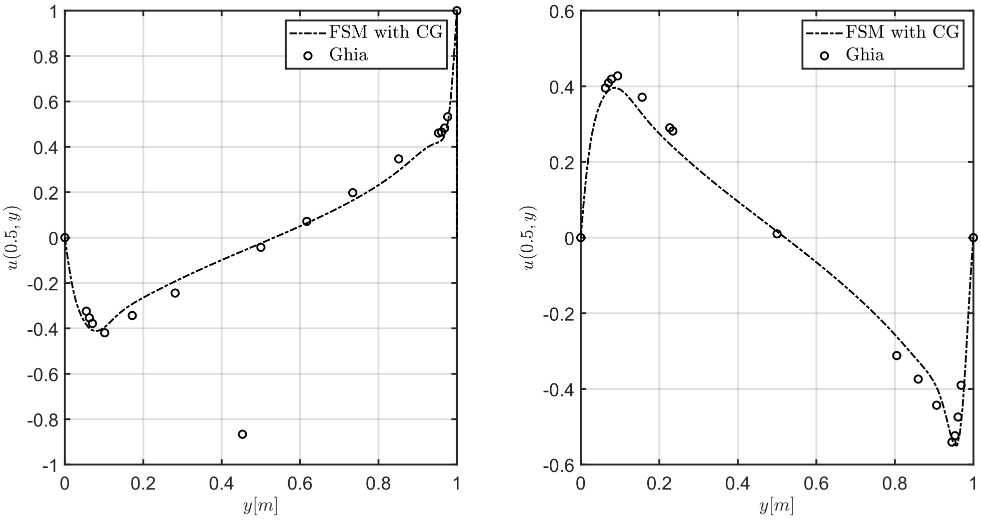


Figure C.12: Comparison between the results at the specific points calculated using the FSM algorithm using the CG algorithm and the results given by [6] using a 80 × 80 mesh in Re = 3200.

Computational time analysis

The time required per different Re for the same mesh sizes is quite similar, with differences getting bigger as the Reynolds number increases, in the case of the FSM, as it can be seen in Figure C.13. Even though, the higher Reynolds is, at each and very mesh configuration, the most expensive solution, as its complexity gets increased.

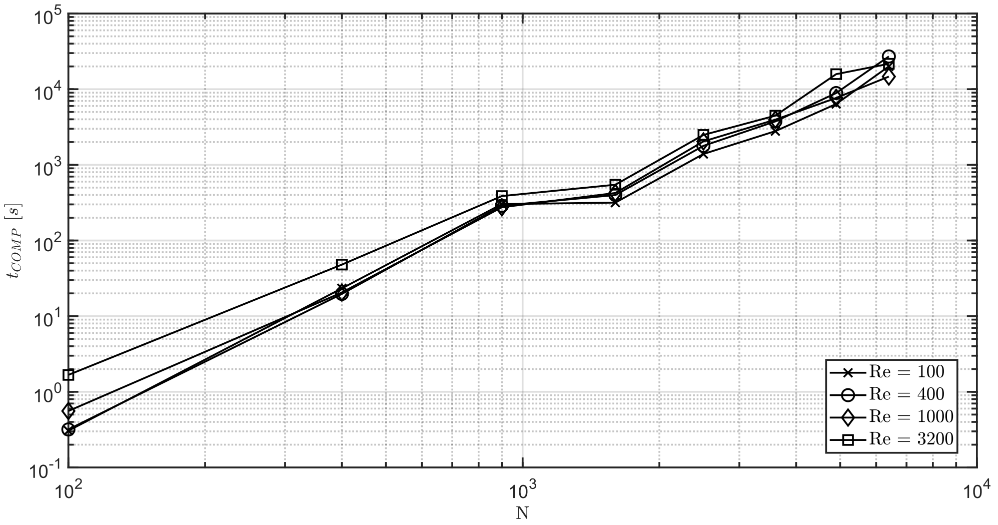


Figure C.13: Computational time spent on the resolution of each different Re for this problem.

C.4 Conclusion

This report was written to present the development of both SIMPLER and FSM algorithms applied to the Lid-Driven Cavity, which is one of the basic problems in the Computational Fluid Dynamics.

The results obtained using the SIMPLER algorithm solved with a LBL scheme suggest that the development of the code has been correct, with a mesh refinement sufficiently elevated which provide solid results when compared to the ones obtained in [6]. Furthermore, the use of the full cosine distribution of the mesh implies that a lower number of control volumes are required to obtain plausible results, since the difference between the resultd obtained by the original code and the results extracted from [6] are quite similar even if the number of control volumes used is quite lower (80 for 129).

Nevertheless, the use of LBL was too slow for higher-Reynolds simulations, which led to the implementation of Conjugate Gradient, a more efficient solver which is capable of being parallelized, being this one of the reasons why it is commonly used in any kind of numerical methods. At this point, the velocity of the simulations was increased and, combined with the higher timestep that SIMPLER allows due to its semi-implicit properties, the resolution was faster.

What outstands from the latter is the fact that the calculation time at different mesh sizes is approximately constant, which implies that, given a mesh, the time required for Re = 100 or Re = 1000 does not differ a lot. This ”constant” resolution time has its sense in the fact that a constant timestep as provided by the CFL condition at each iteration will make the computational time similar.

It is important to remark that in some of the reference results in [6] an error does appear, as happens in Re = 400 or Re = 3200, where some points do not follow the natural tendency of the fluid. These points are not taken in consideration.

Hence, the basic conclusions which can be extracted from the results is that, for lower Reynolds numbers, SIMPLER is generally faster, since the physical timestep (defined as the time scale of the fluid motion) is big enough so that with wider timesteps as SIMPLER allows, its convergence onto valid results is faster.

On the other hand, at slightly higher Reynolds numbers, insofar as its value is closer to the transition from laminar to turbulent inside of a duct (which is generally tabulated at approximately 2900), yet also for Re = 1000 the FSM becomes faster, due to timestep physical size, which becomes smaller than the latter case, being the variable timestep set by the CFL condition more accurate to catch these movement that the bigger timesteps used in the semi-implicit method.

C.5 Results

C.5.1 SIMPLER

Re = 100

Grid size dependance

N	$u(0.50, 0.50)$	$v(0.50, 0.50)$	ε_u	ε_v	$\varepsilon_{G,u}$	$\varepsilon_{G,v}$	t_{comp} (s)
100	-0.1452	0.0330	0.2965	0.3891	0.2944	0.3951	2.57
400	-0.1778	0.0409	0.1389	0.2431	0.1363	0.2506	23.32
900	-0.1912	0.0456	0.0739	0.1554	0.0711	0.1637	86.27
1600	-0.1976	0.0485	0.0430	0.1026	0.0401	0.1115	252.41
2500	-0.2010	0.0503	0.0263	0.0689	0.0233	0.0781	491.19
3600	-0.2031	0.0515	0.0164	0.0460	0.0133	0.0554	1155.01
4900	-0.2034	0.0524	0.0148	0.0295	0.0117	0.0391	2363.44
6400	-0.2053	0.0531	0.0055	0.0172	0.0024	0.0268	4321.65
8100	-0.2060	0.0536	0.0023	0.0076	0.0007	0.0174	6696.65
10000	-0.2064	0.0540	-	-	0.0031	0.0098	9544.33

Table C.1: Convergence analysis results for Re = 100, a relaxation factor $f_r = 0.90$, a time-variation tolerance of 10^{-7} and $\Delta_t = 0.01$ s, where ε_G stands for the difference with Ghia [6] reference results for both u and v.

Timestep analysis

Δ_t (s)	$u(0.50, 0.50)$	$v(0.50, 0.50)$	$\varepsilon_{u,min}$	$\varepsilon_{v,min}$	t_{sim} (s)	t_{comp} (s)
10	-0.201013	0.05028	0.000502708	0.000435371	140	232.523
1	-0.201013	0.05028	0.000502708	0.000435371	37	285.204
0.5	-0.201013	0.05028	0.000502708	0.000435371	29.5	294.73
0.1	-0.201012	0.0502801	0.00049773	0.000433383	22.3	399.777
0.05	-0.201012	0.0502802	0.00049773	0.000431395	20.6	434.119
0.01	-0.201008	0.0502811	0.000477821	0.000413503	17.29	491.191
0.005	-0.201003	0.0502822	0.000452935	0.000391635	16	771.204
0.001	-0.200962	0.050291	0.000248865	0.000216692	13.044	2063.22
0.0005	-0.200912	0.0503019	-	-	11.789	3306.46

Table C.2: Time discretization convergence analysis results for Re = 100, a relaxation factor $f_r = 0.90$, a time-variation tolerance of 10^{-7} and 2500 control volumes.

Relaxation analysis

f_r	$N = 1600$	$N = 2500$	$N = 3600$
0.70	217.331	484.947	991.414
0.80	206.874	472.207	989.519
0.90	200.585	468.740	986.017
1.00	198.348	469.865	999.080
1.10	201.086	467.863	1023.860
1.20	207.753	479.654	1143.670

Table C.3: Influence of the relaxation factor por different number of control volumes per a timestep $\Delta_t = 0.01\ s$ in Re $= 100$.

Tolerance analysis

δ	$u(0.50, 0.50)$	$v(0.50, 0.50)$	$\varepsilon_{u,10^{-10}}$	$\varepsilon_{v,10^{-10}}$	$t_{sim}\ (s)$
10^{-2}	-0.048662	0.001219	0.757914	0.975749	28.50
10^{-3}	-0.137663	0.024801	0.315154	0.506742	121.56
10^{-4}	-0.195489	0.050554	0.027481	0.005442	280.01
10^{-5}	-0.200512	0.050382	0.002492	0.002029	416.61
10^{-6}	-0.200963	0.050291	0.000249	0.000209	520.06
10^{-7}	-0.201008	0.050281	2.4874×10^{-5}	2.18775×10^{-5}	621.57
10^{-8}	-0.201012	0.050280	4.9748×10^{-6}	3.97772×10^{-6}	637.08
10^{-10}	-0.201013	0.050280	-	-	767.46

Table C.4: Tolerance analysis results for Re $= 100$, a relaxation factor $f_r = 0.90$, a timestep $\Delta_t = 0.01\ s$ and 2500 control volumes.

Re = 400

Grid size dependance

N	$u(0.50, 0.50)$	$v(0.50, 0.50)$	ε_u	ε_v	$\varepsilon_{G,u}$	$\varepsilon_{G,v}$	$t_{comp}\ (s)$
100	-0.0919	0.0627	0.2631	0.0099	0.1994	0.2091	7.05
400	-0.1293	0.0923	0.0374	0.4576	0.1270	0.7801	59.25
900	-0.1365	0.0895	0.0950	0.4134	0.1896	0.7261	206.93
1600	-0.1350	0.0820	0.0825	0.2949	0.1760	0.5813	458.36
2500	-0.1324	0.0761	0.0619	0.2015	0.1536	0.4673	916.20
3600	-0.1301	0.0519	0.0438	0.1813	0.1340	0.0001	1695.70
4900	-0.1283	0.0688	0.0292	0.0860	0.1181	0.3263	2939.41
6400	-0.1269	0.0664	0.0174	0.0492	0.1053	0.2813	4824.95
8100	-0.1257	0.0647	0.0079	0.0220	0.0949	0.2480	9502.09
10000	-0.1247	0.0633	-	-	0.0864	0.2212	14682.80

Table C.5: Convergence analysis results for Re $= 400$, a relaxation factor $f_r = 0.90$, a time-variation tolerance of 10^{-7} and $\Delta_t = 0.01\ s$, where ε_G stands for the difference with Ghia [6] reference results for both u and v.

C.5.2 FSM

Re = 100

Grid size dependance

N	$u(0.50, 0.50)$	$v(0.50, 0.50)$	$\varepsilon_{6400,u}$	$\varepsilon_{6400,v}$	t_{comp} (s)	N. of iterations
100	-0.15656	0.07040	0.21804	0.12752	0.305956	1494
400	-0.18743	0.06868	0.06385	0.09999	23.2854	17758
900	-0.19676	0.06603	0.01727	0.05753	302.703	72914
1600	-0.20058	0.06449	0.00181	0.03293	316.676	192617
2500	-0.20227	0.06368	0.01024	0.01990	1391.4	398457
3600	-0.20274	0.06329	0.01260	0.01361	2788.27	703817
4900	-0.20214	0.06302	0.00962	0.00929	6373.42	1110746
6400	-0.20021	0.06244	-	-	20052.4	1607333

Table C.6: Convergence analysis results for Re = 100, using a relaxation factor of $f = 0.9$, a time variation tolerance of 10^{-6} and variable timestep set by the CFL condition.

Re = 400

Grid Size Dependance

N	$u(0.50, 0.50)$	$v(0.50, 0.50)$	$\varepsilon_{6400,u}$	$\varepsilon_{6400,v}$	t_{comp}	N. of iterations
100	-0.05631	0.07262	0.86540	0.18935	0.31824	1479
400	-0.07300	0.06336	0.43904	0.07081	19.6751	12747
900	-0.08255	0.05839	0.27258	0.00816	288.211	51000
1600	-0.08886	0.06445	0.18223	0.08653	399.044	138196
2500	-0.09356	0.05589	0.12282	0.05331	1782.57	301758
3600	-0.09751	0.05620	0.07727	0.04758	3750.74	572678
4900	-0.10123	0.05718	0.03768	0.02957	8891.02	974727
6400	-0.10505	0.05887	-	-	27095.8	1532319

Table C.7: Convergence analysis results for Re = 400, using a relaxation factor of $f = 0.9$, a time variation tolerance of 10^{-6} and variable timestep set by the CFL condition.

Re = 1000

Grid size dependance

N	$u(0.50, 0.50)$	$v(0.50, 0.50)$	$\varepsilon_{6400,u}$	$\varepsilon_{6400,v}$	t_{comp}	N. of iterations
100	-0.02399	0.03883	0.52890	0.34654	0.557517	2601
400	-0.03475	0.03651	0.31750	0.26621	20.5396	13132
900	-0.04023	0.03290	0.20997	0.14100	274.874	47725
1600	-0.04354	0.03068	0.14488	0.06381	424.593	118080
2500	-0.04550	0.02928	0.10656	0.01549	2031.31	229622
3600	-0.04726	0.02856	0.07187	0.00950	3958	396087
4900	-0.04905	0.02832	0.03667	0.01797	7664.36	638632
6400	-0.05092	0.02884	-	-	14680	1004425

Table C.8: Convergence analysis results for Re = 1000, using a relaxation factor of $f = 0.9$, a time variation tolerance of 10^{-6} and variable timestep set by the CFL condition.

Re = 3200

Grid size dependance

N	$u(0.50, 0.50)$	$v(0.50, 0.50)$	$\varepsilon_{6400,u}$	$\varepsilon_{6400,v}$	t_{comp}	Number of iterations
100	-0.01480	0.02874	0.40375	0.78112	1.67204	7589
400	-0.01579	0.02481	0.36389	0.53763	48.0134	31504
900	-0.01858	0.02087	0.25142	0.29371	386.56	63931
1600	-0.02083	0.01894	0.16072	0.17372	546.059	117472
2500	-0.02227	0.01756	0.10291	0.08808	2483.73	217793
3600	-0.02337	0.01672	0.05835	0.03627	4468.29	389905
4900	-0.02468	0.01606	0.00573	0.00488	15808.4	671327
6400	-0.02482	0.01614	-	-	21660.1	1078018

Table C.9: Convergence analysis results for Re = 3200, using a relaxation factor of $f = 0.9$, a time variation tolerance of 10^{-6} and variable timestep set by the CFL condition.

Appendix D

Differentially Heated Cavity

The purpose of this chapter is to solve a differentialy heated cavity problem using the Fractional Step Method in order to validate a Navier-Stokes laminar solver which has incorporated the energy equation as well as the consideration of mass forces (gravity) using Boussinesq's approximation (C.1).

Where u_i stands for the velocity field in dimension i, which is defined using x_i; ρ_0 stands for the density at a reference temperature T_0, μ for the dynamic viscosity, p for the pressure field, β for the coefficient of volumetric expansion, C_p for the specific heat at constant pressure and λ stands for the thermal conducitivity.

The problem, then can be defined as follows (adapted from [7]):

Consider the two-dimensional flow of a fluid with $Pr = 0.71$ *on am upright square cavity described in terms of* $0 \leq x \leq 1$, $0 \leq y \leq 1$ *with* y *vertically upwards and a Prandtl number of* 0.71*. Assume that both components of the velocity are zero on all the boundaries, that the boundaries at* $z = 0, 1$ *are insulated and that* $T(0, y) = T_1$ *and* $T(1, y) = T_2 = T_1 - \Delta T$.

Calculate the fow and thermal field for Rayleigh numbers (D.1) *of* $10^3, 10^4, 10^5, 10^6$*, supplying the following results:*

- *Average Nusselt number.*
- *Nusselt number at the hot wall and at the vertical mid-plane*
- *Maximum vertical velocity on the horizontal mid-plane and its location.*
- *Maximum horizontal velocity on the vertical mid-plane and its location.*

$$\text{Ra} = \frac{\rho^3 C_p \beta D^3 g \Delta T}{\mu \lambda} \tag{D.1}$$

Then, an outline for the problem can be found in Figure D.1. Whereas the air properties used will be recopilated in Table D.1, which are extracted and interpolated from [31].

Hot temperature (K)	373.15
Density, ρ ($\text{kg} \cdot \text{m}^{-3}$)	0.9458
Specific heat, C_p ($\text{J} \cdot \text{kg}^{-1}\text{K}^{-1}$)	1009
Thermal conductivity, λ ($\text{W} \cdot \text{m}^{-1}\text{K}^{-1}$)	0.03095
Dynamic viscosity, μ ($\text{kg} \cdot \text{m}^{-1}\text{s}^{-1}$)	2.181×10^{-5}

Table D.1: Air properties for Pr = 0.7111 extracted from [31].

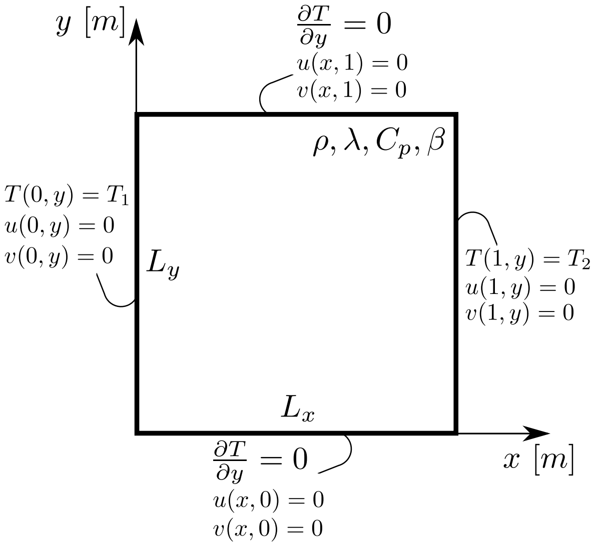


Figure D.1: Outline for the problem. Adapted from [7].

D.1 Evaluation of the convective-diffusive term for temperature

Following the same procedure as for the velocity components, in the end, (D.2) is found.

$$R(T_P) = \frac{1}{\Omega}\left[\frac{\lambda_e \Delta_y}{\Delta_x^{PE}} T_E + \frac{\lambda_w \Delta_y}{\Delta_x^{PW}} T_W + \frac{\lambda_n \Delta_x}{\Delta_y^{PN}} T_N + \frac{\lambda_s \Delta_x}{\Delta_y^{PS}} T_S - \left(\frac{\lambda_e \Delta_y}{\Delta_x^{PE}} + \frac{\lambda_w \Delta_y}{\Delta_x^{PW}} + \right.\right. \tag{D.2}$$

$$\left.\left. + \frac{\lambda_n \Delta_x}{\Delta_y^{PN}} + \frac{\lambda_s \Delta_x}{\Delta_y^{PS}}\right) T_P + \rho C_p (\Delta_y (u_w T_w - u_e T_e) + \Delta_x (v_s T_s - v_n T_n))\right]$$

D.2 Algorithm of resolution

In order to solve the problem, the following algorithm has been developed:

1. **Data input**: physical and numerical data
2. **Basic and preliminary calculations**, corresponding to the mesh, the properties, the time step and the initial conditions. Furthermore, the first stage R vectors have been also computed. Furthermore, and considering that its value is a constant throughout the problem, the pressure coefficients have also been computed.
3. **Fractional Step Method:** which will be divided into the following steps:
   (a) Calculation of R for both temperature and velocity fields using (D.2), (C.17) and (C.19).
   (b) Calculation of temperature for this time step using (D.2).
   (c) Computation of the predictor velocity field using (C.13).
   (d) Solve the pressure equation using the Conjugate Gradient.
   (e) Correct the predictor velocity in order to obtain the velocity field at this timestep using (C.12).

(f) If convergence has been achieved (D.3), then, move to 4. If not, then, return to (a) with $R^{n-1} = R^n$ and $\vec{u}^{n-1} = \vec{u}^n$.

$$\frac{\max(\delta_u, \delta_v, \delta_T)}{\Delta_t} < \delta \tag{D.3}$$

where $\delta_\xi = \sqrt{\sum_{\forall i,j} (\xi_{ij}^{n+1} - \xi_{ij}^n)^2}$, being ξ a dummy variable for u, v, T

4. **Postprocessing:** calculate dimensionless velocities and temperatures, compute Nusselt number.

5. **End**

D.3 Results

In order to compare the results, both temperature and velocities fields are obtiained. Then, since the deliverables of the problem are the average Nusselt number as well as the maximum and the minimum local Nusselt number at the hot wall. Then, these dimensionless numbers have to be defined.

In order to do so, non-dimensional parameters have to be defined. In the case of velocities, they will be given their dimensionless velocity, which corresponds to (D.4a). In the case of temperatures, and considering the main dependance is on its gradient, the dimensionless value vill be given by (D.4b).

$$\bar{u}_i = \frac{u_i L_x}{\alpha} = \frac{u_i L_x \rho C_p}{\lambda} \tag{D.4a}$$

$$\theta = \frac{T - T_2}{T_1 - T_2} \tag{D.4b}$$

Then, as stated by [8], the dimensionless heat flux, also known as Nusselt number, in any position could be estimated as (D.5), where $\bar{x} = x/L_x$.

$$\mathrm{Nu}_x(x, y) = \bar{u}_i \theta - \frac{\partial \theta}{\partial \bar{x}_i} \tag{D.5}$$

In order to obtain the derivative of θ, three different methods depending on the position of the control volume regarding the x direction have been considered. First of all, for the boundary-neighbour nodes, the temperature field will be approximated using a parabolic approach (D.6) for a given y position, which solution will be available at section D.6.

$$\theta(x) \approx a + bx + cx^2 \tag{D.6}$$

Therefore, the parameter required for (D.5) is its derivative. Thus, its derivative will be used to determine the heat flux.

For its neighbors, a CDS has been applied, considering the derivatives at the boundaries of the control volume, which can be calculated using a first order Taylor-like approximation for both sides. Then, developing the expression, (D.7) will be obtained.

$$\frac{\partial \theta}{\partial \bar{x}} = \frac{\theta_{i+1}(x_i - x_{i-1}) + \theta_i(x_{i+1} - x_{i-1}) + \theta_{i-1}(x_{i+1} - x_i)}{2(x_i - x_{i-1})(x_{i+1} - x_i)} \tag{D.7}$$

Eventually, for all inner nodes, a fourth order interpolation, considered using the a five-point-stencil-like interpolation (D.8), has been performed, so as to find the proper and most accurate values, which has been done after considering that subtracting combinations of the δ values at a power higher than 1 are much smaller than the first power parameters.

$$\frac{\partial\theta}{\partial\bar{x}} = \frac{\theta_{i+1} - \theta_{i-1} - \theta_{i+2} + \theta_{i-2}}{\delta_1 - \delta_2 + \delta_3 - \delta_4} \tag{D.8}$$

where:

$$\delta_1 = x_{i+1} - x_i \qquad \delta_3 = x_i - x_{i-1}$$

$$\delta_2 = x_{i+2} - x_i \qquad \delta_4 = x_i - x_{i-2}$$

Then, this could be seen as the Nusselt in any position $(\bar{x}, \bar{y})$. Hence, in order to find the values for any position $\bar{x}$ and its average, (D.9) will be considered.

$$\overline{\mathrm{Nu}_x} = \int_0^1 \mathrm{Nu}_x(x, y)\ dy \tag{D.9a}$$

$$\overline{\mathrm{Nu}} = \int_0^1 \overline{\mathrm{Nu}_x}\ dx \tag{D.9b}$$

D.3.1 Ra $= 10^3$

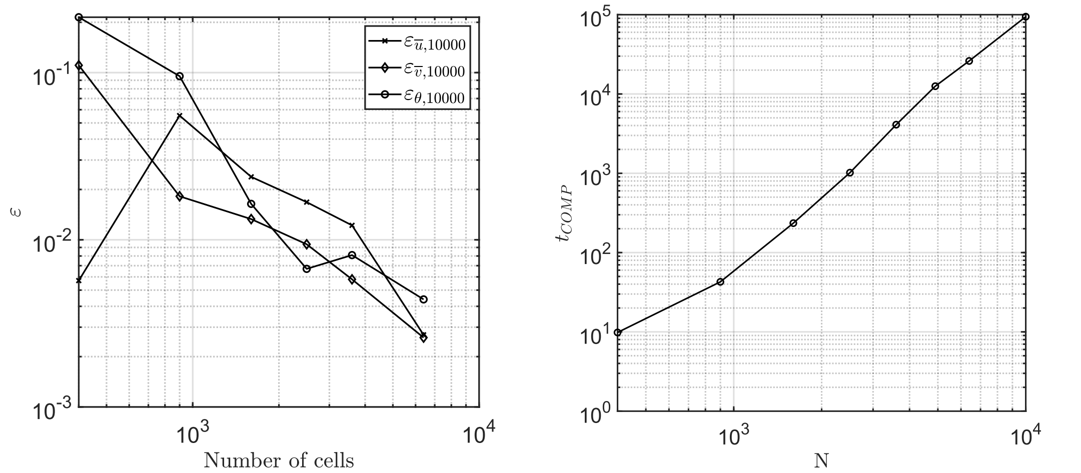
First of all, the calculations for a Rayleigh of 1000 were done. Hence, after applying the methodology previously explained in a general case, the results were obtained, being summarized in Table D.3, regarding temperature, horizontal and vertical speed in the same notation as [8]. The maximum values, nevertheless, differ (difference stabilized at around 5%, value evolution available at Table D.4) from the reference benchmark at [8]. Furthermore, a grid convergence analysis was also performed, as well as an analysis of computational cost in terms of computational time, which results are available in Table D.4.

Regarding the convergence of the method, and as it can be seen in Figure D.2a, which is calculated as (D.10).

$$\varepsilon_\xi = \left|\frac{\xi - \xi_{10000}}{\xi_{10000}}\right| \tag{D.10}$$

As it can be seen, the behaviour is as expected, with the exception in the horitzontal speed $\overline{u}$, the error of which increases for 900 control volumes. Nevertheless, given the fact that it corresponds to a such a coarse mesh, the validity of the results is negligible, being this the reason why only denser meshes will be considered when doing further analysis.

On the other hand, the computational time (Figure D.2b) follows an almost linear log-log behaviour of order $\mathcal{O}(N^3)$ (2.991), even though Conjugate Gradient should be a $\mathcal{O}(N \log N)$



(a) Convergence plots for velocities and temperature at $\mathrm{Ra} = 10^3$.

(b) Computational time in seconds for $\mathrm{Ra} = 10^3$.

Figure D.2: Results for $\mathrm{Ra} = 10^3$.

Eventually, the Nusselt number has been computed, which results can be seen in Figure D.3, even though the recopilation of results is available at Table D.11. This result indicates the magnitude of the heat transfer due to convection compared to the magnitude of heat transfer due to diffusion.

In order to complement the results in the table, the calculations have also been performed for $N = 10000$, with the results from Table D.2.

$\overline{\mathrm{Nu}}_0$	$\overline{\mathrm{Nu}}_{1/2}$	$\overline{\mathrm{Nu}}$
1.1000	1.1081	1.0960

Table D.2: Nusselt calculations for $N = 10000$ and $\mathrm{Ra} = 10^3$.

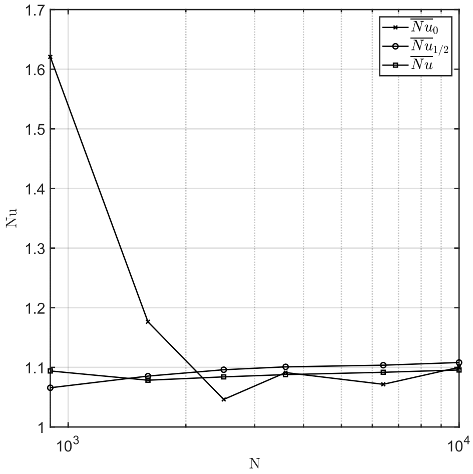


Figure D.3: Nusselt number at the hot wall, at the vertical mid-plane and global average for $\mathrm{Ra} = 10^3$.

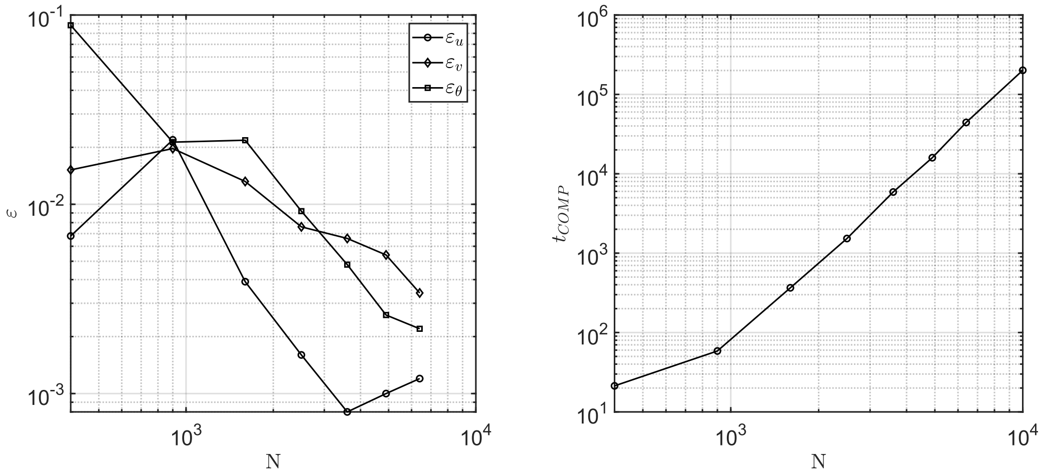
As it can be seen, in this scenario, the heat transfer is equally distributed between both phenomena, convection and conduction, given that its value is around 1. Hence, this is the explanation why the temperature field [9] follows a distribution which is similar to a 1D conduction case (or a 2D conduction with adiabatic top and bottom), with just a slightly deviation of the isotherms, indicating the presence of buoyancy.

D.3.2 Ra $= 10^4$

After obtaining the latter results, the calculations for Ra $= 10^4$ were performed, which results are described in Table D.5. Its velocity fields can be seen in [9]. In this case, the difference between the results and [8] is at around a 3%.

Regarding the convergence of the grid, the same calculations as in the first case can be performed, which result in Figure D.4a, whereas the computational time follows also a $\mathcal{O}(N^3)$ behaviour (2.975), as it can be seen in Figure D.4b.

Finally, the Nusselt number has also been computed, yielding Figure D.5, which result represents that the heat transfer due to convection will be significantly more important than the same phenomena due to conduction.



(a) Convergence plots for velocities and temperature at Ra $= 10^4$.

(b) Computational time in seconds for Ra $= 10^4$.

Figure D.4: Results for Ra $= 10^4$.

D.3.3 Ra $= 10^5$

Following the same exact procedure, the results for Ra $= 10^5$ can be obtained. Considering the grid convergence, Figure D.6a was obtained, showing that the grids should be much more denser in order to obtain pretty good converged results.

On the other hand, the time complexity of the case follows an equivalent computational cost as the results previously presented (Figure D.2b,D.4b), yet with higher values given the higher complexity of the case, yielding Figure D.6b.

Finally, the Nusselt number has also been computed, yielding a greater convection than diffusion weight in the heat transfer process, as it was expected due to its higher Rayleigh number. Its behaviour and convergence can be seen at Figure D.7.

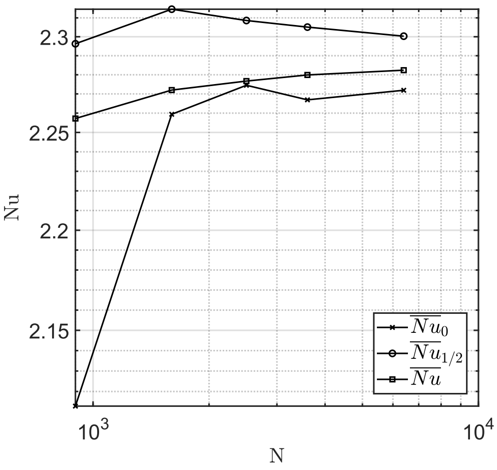
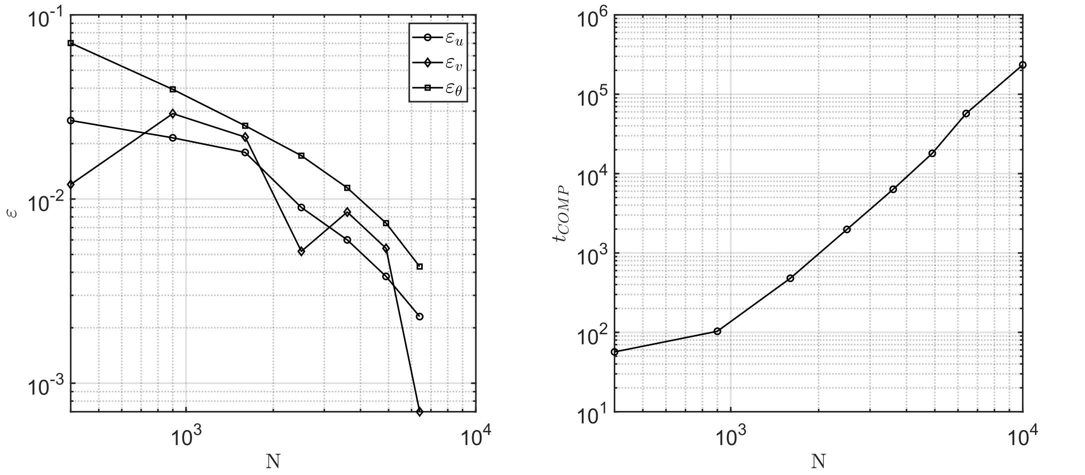


Figure D.5: Nusselt number at the hot wall, at the vertical mid-plane and global average for $\mathrm{Ra} = 10^4$



(a) Convergence plots for velocities and temperature at $\mathrm{Ra} = 10^5$.

(b) Computational time in seconds for $\mathrm{Ra} = 10^5$.

Figure D.6: Results for $\mathrm{Ra} = 10^5$.

D.3.4 **Ra** $= 10^6$

Eventually, the same procedure was developed for $\mathrm{Ra} = 10^6$, which yield Figure D.8a for the grid convergence and Figure D.8b in terms of computational cost, which, in fact, follow similar tendencies as the latter values.

Finally the importance of convection regarding diffusion has also been studied, yielding Figure D.9.

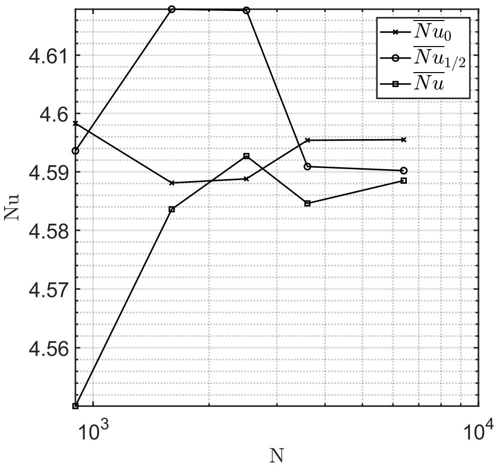
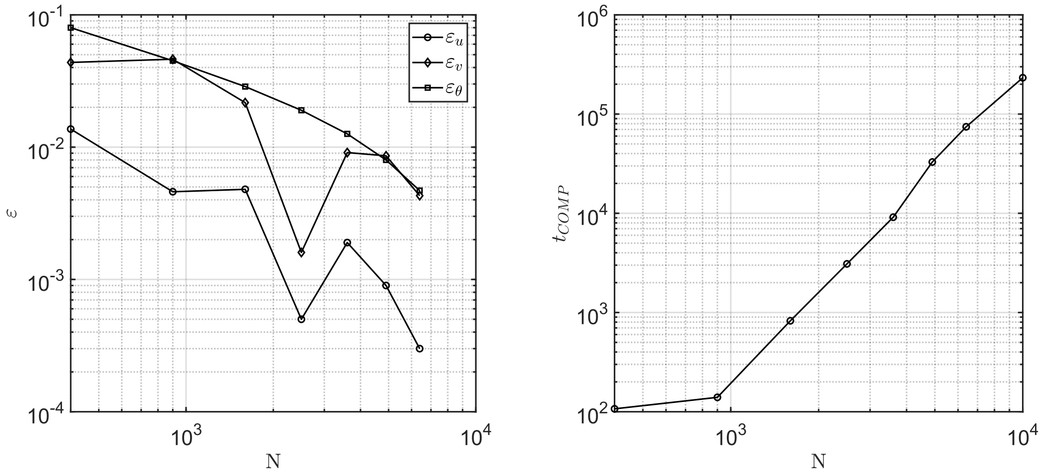


Figure D.7: Nusselt number at the hot wall, at the vertical mid-plane and global average for Ra = 10^5



(a) Convergence plots for velocities and temperature at Ra = 10^6.

(b) Computational time in seconds for Ra = 10^6.

Figure D.8: Results for Ra = 10^6.

D.4 Conclusions

In this report, the solution and comparison with a benchmark solution [8] of a Differentially Heated Cavity problem is intended. In order to do so, a Finite-Volume Discretization (FVM) has been developed, at which the Fractional Step Method (FSM) is applied, following the trend developed in [5], solving the system of equations resulting of the discretization using a Conjugate Gradient.

The main challenge in this problem was the introduction of the buoyancy factor in the momentum equation, which is computed using Boussinesq's approximation; as well as the introduction of the energy equation, all of them developed in (C.1). Furthermore, the use of realistic physical properties also required the interpolation of different factors in order to provide a solid starting point for the program.

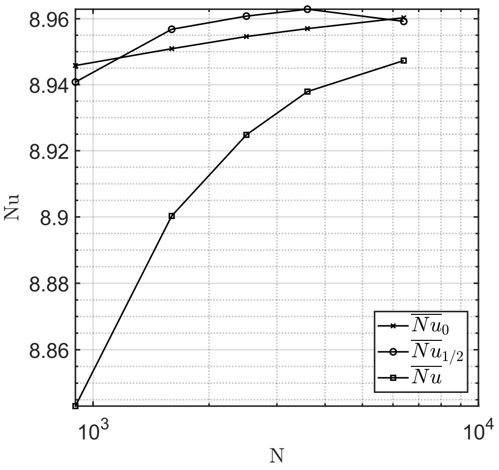


Figure D.9: Nusselt number at the hot wall, at the vertical mid-plane and global average for $\mathrm{Ra} = 10^6$

Furthermore, the use of the L2-norm in calculating the error in all magnitudes, as well as the division by the timestep used, made the convergence of the program much complex, but much more stable in terms of results obtained in comparison with some other simpler methods such as an absolute or relative error, which only take in consideration the maximum deviation of the process.

After performing all corresponding calculations, different conclusions can be extracted from the results.

First of all, regarding the flow behaviour and the heat transfer phenomenology, it can be seen that for $\mathrm{Ra} = 10^3$, the dominant heat transfer phenomena was heat diffusion, as the temperature distribution follows clearly a diffusive pattern (see any conduction case with adiabatic top and bottom boundaries, which can be approximated to a 1D-conduction-case). Nonetheless, regarding all other cases studied, the dominant heat transfer method was convection, in which the temperature boundary layer is more determined for higher Rayleigh numbers [9]. This behaviour is also explained by the Nusselt number assigned onto each and every case, which results can be recalled at Table D.11.

Moreover, in regards to the convergence of the results, even though it is quite good for the mesh sizes developed in this report, a deeper analysis with denser grids should be developed in order to obtain exactly the results expected and proposed in the reference document [7]. Hence, the results obtained in this case are not precisely coherent with the results proposed in [8], yet are good enough to be considered as acceptable. This can be explained, among other possible reasons, by the use of slightly different properties, given that the actual Prandtl number used corresponds to 0.7111, instead of 0.71; which, in fact, determine whether the dimensionless magnitudes correspond, or not, to the benchmark solution.

Furthermore, the convergence within the code follows an expected trend, being this the reason why the results can be ensured to be good for denser grids. Furthermore, it is important to remark that the grid convergence would require denser grids for higher Rayleigh numbers, which in fact is shown when considering the errors within the most dense grid studied in this report.

In regards to the computational complexity of the problem, it has been seen that for a higher Rayleigh number, the time spent in reaching the steady state situation increases following an almost $\mathcal{O}(N^3)$ in all cases, being this a descending number for an increasing Rayleigh.

Furthermore, an almost linear behaviour in log-log plots can be obtained, as showed in Figure D.10.

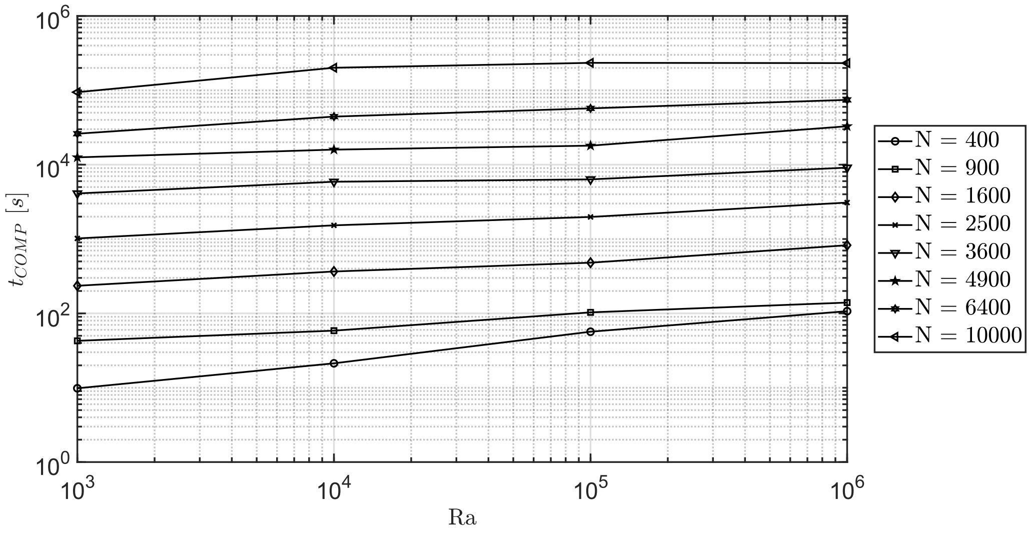


Figure D.10: Computational time as a function of the mesh used as well as the Rayleigh number.

This yields a behaviour in which (D.11) is deduced, for the code used in the solution of this problem, without considering the mesh size, which scales the actual time consumed, being this an absolutely empirical value.

$$t_{COMP} \sim \mathcal{O}(\mathrm{Ra}^{0.1717}) \tag{D.11}$$

D.5 Numerical results

Recall that the benchmark positions for maximum dimensionsless velocities specified at the tables differ from [8], given that the hot and cold walls are placed differently in this report. Furthermore, the results for Ra = $10^4, 10^5, 10^6$ are calculated for a Pr = 0.7111, being this the reason why they differ from the benchmark solution. Nonetheless, in order to verify the code, Ra = 10^3 is calculated using a Pr = 0.71.

N	$\overline{u}_{max}(0.50, y)$	$y_{u,max}$	$\overline{v}_{max}(x, 0.50)$	$x_{v,max}$	$\theta(0.50, 0.50)$
400	3.6399	0.1763	4.1182	0.8237	0.6020
900	3.4582	0.1858	3.6408	0.8531	0.5429
1600	3.5735	0.1907	3.6587	0.8391	0.5038
2500	3.5992	0.1937	3.6735	0.8305	0.4990
3600	3.6160	0.1754	3.6867	0.8246	0.4997
6400	3.6508	0.1829	3.6985	0.8171	0.4935
10000	3.6607	0.1874	3.7082	0.8247	0.4957
Benchmark	**3.649**	**0.187**	**3.697**	**0.822**	-

Table D.3: Numerical results for Ra = 10^3.

N	$\varepsilon_{\overline{u},10000}$	$\varepsilon_{\overline{v},10000}$	$\varepsilon_{\overline{u},DAVIES}$	$\varepsilon_{\overline{v},DAVIES}$	ε_{θ}	t_{COMP} (s)	t_{PHYS} (s)
400	0.0057	0.1106	0.0025	0.1139	0.2145	9.83	4543.09
900	0.0553	0.0182	0.0523	0.0152	0.0952	42.74	3107.91
1600	0.0238	0.0133	0.0207	0.0104	0.0164	235.11	3445.84
2500	0.0168	0.0094	0.0137	0.0064	0.0067	1018.42	3613.45
3600	0.0122	0.0058	0.0090	0.0028	0.0081	4102.52	3745.69
6400	0.0027	0.0026	0.0005	0.0004	0.0044	26092.40	3933.60
10000	-	-	0.0032	0.0030	-	94385.10	4072.40

Table D.4: Computational performance for Ra = 10^3.

N	$\overline{u}_{max}(0.50, y)$	$y_{u,max}$	$\overline{v}_{max}(x, 0.50)$	$x_{v,max}$	$\theta(0.50, 0.50)$
400	16.5891	0.1763	20.8517	0.8790	0.4510
900	16.3378	0.1858	20.9440	0.8880	0.4842
1600	16.6369	0.1609	20.8111	0.8924	0.4840
2500	16.6764	0.1695	20.6948	0.8949	0.4902
3600	16.6889	0.1754	20.6750	0.8801	0.4923
4900	16.6857	0.1796	20.6495	0.8837	0.4934
6400	16.6821	0.1679	20.6082	0.8864	0.4936
10000	16.7028	0.1753	20.5393	0.8802	0.4947
Benchmark	**16.178**	**0.177**	**19.617**	**0.881**	-

Table D.5: Numerical results for Ra = 10^4 at $(0.50, 0.50)$.

D.6 Parabolic approach equations

Given the wall-boundary point as well as its two horizontally speaking closer points, (D.12) will be obtained.

$$\begin{pmatrix} 1 & x_i & x_i^2 \\ 1 & x_{i+1} & x_{i+1}^2 \\ 1 & x_{i+2} & x_{i+2}^2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \theta_i \\ \theta_{i+1} \\ \theta_{i+2} \end{pmatrix} \tag{D.12}$$

By using MATLAB symbolic package, and assigning $x_i = A$, $x_{i+1} = B$, $x_{i+2} = C$,$\theta_i = X$, $\theta_{i+1} = Y$ and $\theta_{i+2} = Z$; it has been solved, yielding (D.13)

$$a = -\frac{ZAB(B-A) + YAC(A-C) + XBC(C-B)}{(A-B)(A-C)(B-C)} \tag{D.13a}$$

$$b = \frac{A^2(Y-Z) + B^2(Z-X) + C^2(X-Y)}{(A-B)(A-C)(B-C)} \tag{D.13b}$$

$$c = -\frac{X(C-B) + Y(A-C) + Z(B-A)}{(A-B)(A-C)(B-C)} \tag{D.13c}$$

N	$\varepsilon_{\overline{u},10000}$	$\varepsilon_{\overline{v},10000}$	$\varepsilon_{\overline{u},DAVIES}$	$\varepsilon_{\overline{v},DAVIES}$	ε_{θ}	t_{COMP} (s)	t_{PHYS} (s)
400	0.0068	0.0152	0.0254	0.0629	0.0883	21.29	9814.35
900	0.0219	0.0197	0.0099	0.0676	0.0213	58.57	3573.76
1600	0.0039	0.0132	0.0284	0.0609	0.0218	365.94	4603.16
2500	0.0016	0.0076	0.0308	0.0549	0.0092	1528.54	4676.92
3600	0.0008	0.0066	0.0316	0.0539	0.0048	5892.41	4723.18
4900	0.0010	0.0054	0.0314	0.0526	0.0026	15940.60	4762.08
6400	0.0012	0.0034	0.0312	0.0505	0.0022	44323.00	4813.98
10000	-	-	0.0324	0.0470	-	201231.00	4859.74

Table D.6: Computational performance for Ra $= 10^4$.

N	$\overline{u}_{max}(0.50, y)$	$y_{u,max}$	$\overline{v}_{max}(x, 0.50)$	$x_{v,max}$	$\theta(0.50, 0.50)$
400	36.5068	0.1210	72.2937	0.9250	0.4569
900	36.3233	0.1469	73.5182	0.9449	0.4721
1600	36.1952	0.1331	72.9889	0.9359	0.4792
2500	35.8784	0.1466	71.8086	0.9302	0.4830
3600	35.7698	0.1374	72.0434	0.9393	0.4858
4900	35.6932	0.1465	71.8242	0.9348	0.4878
6400	35.6411	0.1396	71.4848	0.9409	0.4894
10000	35.5578	0.1410	71.4377	0.9343	0.4915
Benchmark	**34.730**	**0.145**	**68.590**	**0.934**	-

Table D.7: Numerical results for Ra $= 10^5$ at $(0.50, 0.50)$.

N	$\varepsilon_{\overline{u},10000}$	$\varepsilon_{\overline{v},10000}$	$\varepsilon_{\overline{u},DAVIES}$	$\varepsilon_{\overline{v},DAVIES}$	ε_{θ}	t_{COMP} (s)	t_{PHYS} (s)
400	0.0267	0.0120	0.0512	0.0540	0.0703	56.69	26459.40
900	0.0215	0.0291	0.0459	0.0719	0.0394	103.19	4453.55
1600	0.0179	0.0217	0.0422	0.0641	0.0250	480.39	3959.05
2500	0.0090	0.0052	0.0331	0.0469	0.0172	1981.36	4465.13
3600	0.0060	0.0085	0.0299	0.0503	0.0115	6342.50	4478.49
4900	0.0038	0.0054	0.0277	0.0472	0.0074	18012.80	4489.37
6400	0.0023	0.0007	0.0262	0.0422	0.0043	57195.50	4499.62
10000	-	-	0.0238	0.0415	-	234164.00	4513.72

Table D.8: Computational performance for Ra $= 10^5$.

N	$\overline{u}_{max}(0.50, y)$	$y_{u,max}$	$\overline{v}_{max}(x, 0.50)$	$x_{v,max}$	$\theta(0.50, 0.50)$
400	66.9982	0.1210	239.7890	0.9605	0.4515
900	65.7943	0.1120	240.3410	0.9662	0.4687
1600	66.4097	0.1331	234.7240	0.9687	0.4767
2500	66.0594	0.1466	230.1050	0.9702	0.4815
3600	66.2216	0.1374	231.8270	0.9618	0.4846
4900	66.1574	0.1465	231.7060	0.9639	0.4869
6400	66.1145	0.1396	230.7360	0.9655	0.4885
10000	66.0957	0.1410	229.7380	0.9619	0.4908
Benchmark	**64.630**	**0.150**	**219.360**	**0.9621**	-

Table D.9: Numerical results for Ra $= 10^6$ at $(0.50, 0.50)$.

N	$\varepsilon_{\overline{u},10000}$	$\varepsilon_{\overline{v},10000}$	$\varepsilon_{\overline{u},DAVIES}$	$\varepsilon_{\overline{v},DAVIES}$	ε_{θ}	t_{COMP} (s)	t_{PHYS} (s)
400	0.0137	0.0437	0.0366	0.0931	0.0800	107.03	51254.60
900	0.0046	0.0462	0.0180	0.0956	0.0450	139.72	3586.71
1600	0.0048	0.0217	0.0275	0.0700	0.0287	825.22	3728.34
2500	0.0005	0.0016	0.0221	0.0490	0.0190	3091.40	3731.98
3600	0.0019	0.0091	0.0246	0.0568	0.0126	9126.99	3736.11
4900	0.0009	0.0086	0.0236	0.0563	0.0080	32866.20	3741.27
6400	0.0003	0.0043	0.0230	0.0519	0.0047	74433.70	3905.62
10000	-	-	0.0227	0.0473	-	232321.00	4060.18

Table D.10: Computational performance for Ra = 10^6.

	Ra = 10^3			Ra = 10^4		
N	$\overline{\text{Nu}}_0$	$\overline{\text{Nu}}_{1/2}$	$\overline{\text{Nu}}$	$\overline{\text{Nu}}_0$	$\overline{\text{Nu}}_{1/2}$	$\overline{\text{Nu}}$
900	1.6207	1.0658	1.0940	2.1129	2.2963	2.2572
1600	1.1763	1.0853	1.0785	2.2594	2.3146	2.2720
2500	1.0462	1.0960	1.0840	2.2745	2.3086	2.2767
3600	1.0912	1.1009	1.0878	2.2669	2.3051	2.2799
6400	1.0758	1.1036	1.0917	2.2719	2.3003	2.2824
Benchmark	**1.117**	**1.118**	**1.118**	**2.238**	**2.243**	**2.243**

	Ra = 10^5			Ra = 10^5		
N	$\overline{\text{Nu}}_0$	$\overline{\text{Nu}}_{1/2}$	$\overline{\text{Nu}}$	$\overline{\text{Nu}}_0$	$\overline{\text{Nu}}_{1/2}$	$\overline{\text{Nu}}$
900	4.5983	4.5936	4.5501	8.9458	8.9408	8.8431
1600	4.5881	4.6179	4.5836	8.9509	8.9568	8.9003
2500	4.5888	4.6177	4.5927	8.9546	8.9608	8.9248
3600	4.5954	4.5909	4.5846	8.9570	8.9629	8.9379
6400	4.5955	4.5902	4.5885	8.9603	8.9592	8.9473
Benchmark	**4.509**	**4.519**	**4.519**	**8.817**	**8.799**	**8.800**

Table D.11: Nusselt number at the hot wall, at the midplane and the average on the domain for different Rayleighs and different mesh sizes.