POLITECNICO DI TORINO

ENGINEERING SCHOOL
Bachelor's degree in Aerospace Engineering

Thesis

Computational Study of the Numerical Effects caused by the Mesh Quality in a Square Cylinder

Advisors
Domenic D'Ambrosio
Luca Bruno

Author
Almudena López Sanfeliciano

June 2017

Summary

The present Thesis is focused on the quantitative demostration of the effects that a low quality mesh has in the solution of a computational simulation. *Computational Fluid Dynamics* is a recent and complex science, which allows us to approach a numerical result of problems without solution until then. Some advantages are: the wide range of application, its lower cost in comparison with the experimental process, the non-existent limit to the structural dimensions of what we would like to simulate and the time of resolution that is related to the computer, among others. Nonetheless, some disadvantages exist too, like the ease of making a mistake during the setting up of the parameters. The program will always resolve the problem if the solution is converged. So, it is very important to verify the appropiate definition of all parameters and variables of our problem in order to get a suitable solution.

There are many types of errors with several effects in the solution. In this Thesis, the effect of the mesh quality is investigated because of its consequencies in the result. According to the characteristics that we impose to the mesh, it will be associated with a quality. We are going to simulate a fluid flow around a square cylinder, defining four different types of grids for the zone around the cylinder and keeping the rest constant. So, each grid has a concret value of mesh quality. After the simulation, we compare the results of the various cases between them. It has been demostrated that the error propagated by an inadequate definition of the parameters of the grid quality is appreciable. The real magnitude of this effect is relative to the force modulus and its application. For example, in the field of Civil Engineering, the forces that support the structure of a building have a high value and an error in the calculation could have disastrous consequences. For this reason, it should be taken into account, to avoid major damages.

The outline of this Thesis gives a concret idea what each chapter is about and shows schematically the evolutionary process which has taken place during its development.

Chapter 1 introduces the field in which the study is taken place. *Fluid Dynamics* has been presented and its wide range of application in the computational frame. Concretly, the problem of the square cylinder is explained from an aerodynamic point of view.

Chapter 2 is about two articles in which some CFD simulations and experimental data of the fluid flow aroud a square cylinder are explained and compared, giving us some references.

Chapter 3 deals with the methodology followed in a simulation of CFD. The funda-

mental concepts of fluid dynamics are also included and explained in detail to understand the theoretical part of a simulation.

Chapter 4 describes the procedure followed to set up the problem characteristics in the program and run the calculation. Also, it is presented the mesh configuration of the entire domain and the four cases of study.

Chapter 5 collects all results obtained after the simulation. They are compared at different sections, according to the variable studied. Some numerical differences between the solutions of the four types of grids are showed, verifying the effects of the mesh quality.

Finally, **Chapter 6** summarizes the principal conclusions achived after analysing the solutions of the different cases. In addition, some opinions about future perspectives are commented to extend the project towards new directions.

In conclusion, this Thesis provides a wide study about the main fundaments of the *Computational Fluid Dynamics* simulations and, in particular, a deep investigation of the principal numerical effects caused by the mesh quality.

Contents

List of Figures

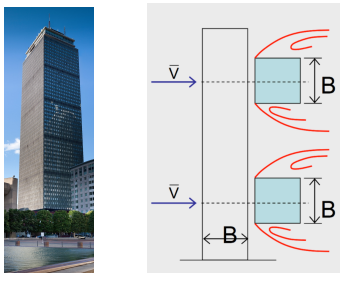
List of Tables

Chapter 1

Introduction

In physics, *Fluid Mechanics* is a discipline that studies the laws of the flow movement and its interaction with the solid bodies. When it talks about *fluids*, it includes *liquids*, *gases* and *plasmas*. *Fluid Mechanics* can be divided into two branches: *Fluid Statics*, which is the study of fluids at rest, and *Fluid Dynamics*, which studies the effect of forces and moments on fluid motion. In addition, it has a lot of interesting applications, including Mechanical Engineering, Civil Engineering, Chemical Engineering, Geophysics, Astrophysics and Biology, among others. Specially, *Fluid Dynamics* has many problems partially or totally unsolved. For this reason, it is an active field of research, in which a modern discipline, known as *Computational Fluid Dynamics (CFD)*, is used to approximate the numerical solution of these problems and compared with experimental methods. Nevertheless, not always experimental data can be obtained because of the budget or in some cases, for example in Civil Engineering, it is complex to carry out an experimental process. In these cases, it is extremly important to obtain very accurate results from CFD simulation.

In this Thesis, the main objective is to go deeper into the error propagated when a mesh with low quality is used. We study the behaviour of a flow around a square cylinder using grids with different mesh quality and then, we compare the results. However, first of all, we should learn about the physical problem (interation between the fluid and the square cylinder) to know what kind of results should obtain and, later, we could define and simulate it.



(a) (b)

Figure 1.1: Skyscraper cross-section

The square cylinder is the great interest in the Civil Engineering because of its similarity with the cross-section of skyscrapers (Fig. **??**). In aerodynamics, the square cylinder is classified as a type of section called *bluff body*. At sufficiently high Reynolds numbers, these bodies cause an early separation of the boundary layer, a thick unsteady vortical wake and time-dependent aerodynamic forces. So, the main contribution to the drag force is due to the pressure forces. The reattachment of the flow depends on the value of the ratio between the length and the height of the section, B/D. Experimentally, the reattachment approximatelly occurs when $B/D \geq 2.8$. Furthermore, as we can see in the figure **??**, there are two more types of sections from an aerodynamic point of view: *semi-streamlined* and *streamlined* section. The last one is the opposite of the *bluff body*. It has the boundary layer totally attached, a thin wake and steady aerodynamic forces. In this case, the principal contribution to the drag forces is the viscous effect of the fluid. The *semi-streamlined* section is a mix of the two previous sections, *bluff* and *streamlined*.

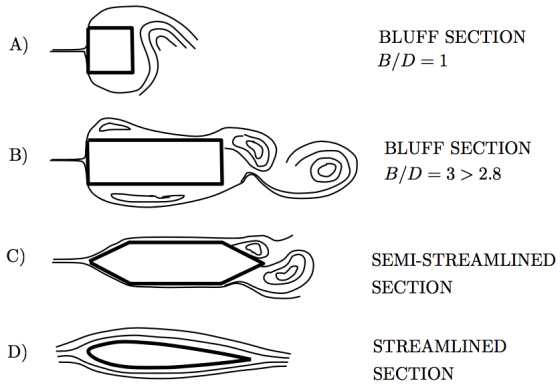


Figure 1.2: Aerodynamic classification of sections

Moreover, the square cylinder has a characteristic flow topology, as it is shown in the figure **??**. In the present figure, we can differentiate three main regions: *front*, *lateral* and *base* region.

- *Frontal region*

  The *stagnation point* is set up in this region. It means the flow does not have velocity at this point. Near the frontal wall, the behaviour of the flow is influenced by the viscous and turbulent stresses.

- *Lateral region*

  On the one hand, in the near lateral region, the boundary layer is separated at the frontal corners of the square cylinder. As a consequence, they are called *boundary layer separation points*. The transition from laminar to turbulent regime takes place in this part. Then, due to the important effects of viscosity, the flow becomes rotational and in some cases, a reattatchment point could appear. This region is significant because it is where the inestability is originated. On the other hand, in the far lateral region, where the shear stresses are not so significant, the flow is laminar and irrotational.

- *Base region*

  The vortical wake is located in this region, where the flow is moved by convective and turbulent effects. From an structural point of view, it is very important to simulate correctly this flow region because some variables of interest depend on it.

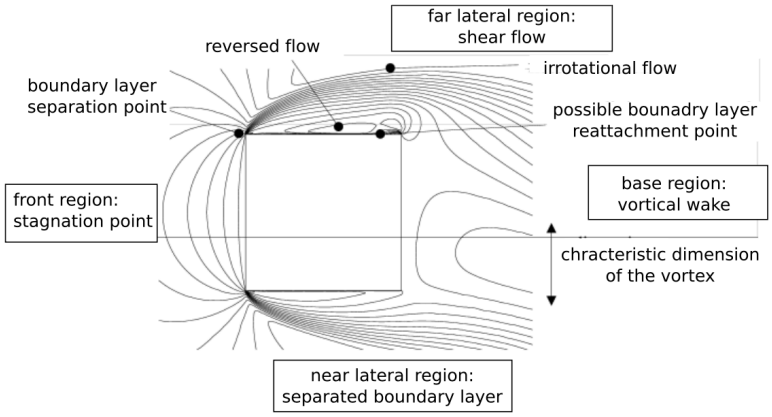


Figure 1.3: Flow topology using horizontal velocity contours

To quantify the complete effects of the flow in the structures, we will calculate the time-averaged and maximum values of the aerodynamic coefficients and the Strouhal number, St. This number, which can be expressed by the dimensionless expression **??**, is used to know the frecuency content of the lift because it is considered to be coincident with the vortex shedding frequency.

$$St = \frac{f_{C_L} D}{U} \tag{1.1}$$

Finally, we should look into the articles and experimental data. In this way, we can verify the theorical concepts and get an idea about the order of magnitud that our results should have.

Chapter 2

Literature Review

The behaviour of the flow around a square cylinder has been analysed deeply in the literature. In this chapter, we are going to summarize some articles related to square cylinder simulations, to understand the essential aspects of this field and to elaborate a rigorous criteria in the determination of the correct results.

2.1 A priori grid quality estimation for high-order finite differencing

In this article, *R. Fattah, D. Angland and X. Zhang* study the correlation between a grid quality metric and the accuracy of the solution.

First of all, they start introducing the problem. As they say, it is very important in *Computational Fluid Dynamics, CFD,* and *Computational Aeroacustics, CAA,* to predict in an accurate way the distortions. Normally, to achieve it *Finite Differencing, FD,* methods are used. However, FD methods generate two types of grid-induced truncation errors because this method is obtained using Taylor series. On the one hand, the first is related to the solution field. On the other hand, the second is attributed to the grid transformation metrics. So, it will be necessary a great *grid quality metric* for the purpose to identify and resolve areas with low quality. There is a similarity between the truncations errors in an spatial scheme, obtained using the Fourier analysis, and the transfer function of the spatial filters.

Then, they explain the methodology to obtain the grid quality metric for a 1D case. Starting from the wave equation in the generalized coordinates, as

$$\frac{\partial f}{\partial t} + U\frac{\partial f}{\partial \xi}\frac{\partial \xi}{\partial x} = 0 \tag{2.1}$$

where f is a scalar variable, t is time, U the convection speed in x-direction and ξ the generalized coordinate.

In Equation **??**, three numerical errors are generated. The temporal derivate depends only on the solution f and it will be discretized using a temporal scheme. The truncation

error in a temporal scheme can be reduced using a smaller time step. The following term, $\frac{\partial f}{\partial \xi}$, will be approximated by a spatial differencing scheme. The truncation error would be greater when the function f is poorly solved along the generalized axis ξ. To decrease this error, the grid should be refined where the variations of the solution f are important. Nevertheless, this refinement only can be carried out if the solution field f is known in advance. Lastly, the term $\frac{\partial \xi}{\partial x}$ is known as the *metric of the grid transformation*. In this case, the trucation error will be larger when the function ξ is not well resolved along the x-coordinate. To reduce it, a mesh refinement should be done but it is not necessary to know the results of the function f beforehand. In this article, it is considered a solution field continuos and a small time step, thereby the most important truncation error is produced by the metrics of the grid transformations. So, the objective of this work will be to define a *grid quality metric* ralated to the trucation errors of $\frac{\partial \xi}{\partial x}$.

Additionally, the grid quality metric is created with the aim of indentifying the regions of the grid where the truncation errors are more important. To approximate the truncation errors are used filters since the similarity between them and they can be defined as,

$$\hat{\triangle}\psi_i = \psi_i - \hat{\psi}_i \tag{2.2}$$

The output from the filter operator give us a rough value of the truncation error produced by the spatial scheme when $\frac{\partial \xi}{\partial x}$ is discretized. Then, the displacement of a point after applying a filter is calculated from the following equation as

$$\hat{\triangle}\xi = \sqrt{\hat{\triangle}x_i^2 + \hat{\triangle}y_i^2 + \hat{\triangle}z_i^2} \tag{2.3}$$

Also, a local length scale is defined to normalize the quantity of the equation **??**

$$\triangle\xi = \sqrt{x_\xi^2 + y_\xi^2 + z_\xi^2} \tag{2.4}$$

Finally, the grid quality metric, Q, is defined in 3D case as

$$Q = \sqrt{\left(\frac{\hat{\triangle}\xi}{\triangle\xi}\right)^2 + \left(\frac{\hat{\triangle}\eta}{\triangle\eta}\right)^2 + \left(\frac{\hat{\triangle}\zeta}{\triangle\zeta}\right)^2} \tag{2.5}$$

where greater values of Q will mean grater is the value of the truncation error and, then, the quality of the grid will be less. The grid quality metric is obtained for all grid points.

Therefore, to prove the correlation among the grid quality metric and the accuracy of the solution, some cases are tested. A simplified problem of an inviscid vortex core convection is used to verify this reciprocity. This problem has been chosen because the analytical solution is known and the errors in the solution can be calculated. To solve it, the two dimensional Euler equations in a conservative form are used.

$$\frac{\partial}{\partial t}\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho e_T \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho(e_T + p)v \end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho(e_T + p)v \end{pmatrix} = 0 \tag{2.6}$$

where ρ, $\vec{u}$, p are the non-dimensional density, velocity and static pressure, respectively, and e_T is the total energy per unit of mass. Temporal term is solved using a fourth-order Runge-Kutta scheme with a Courant number of $CFL < 0.45$ and flux terms are evaluated by fourth-order spatial schemes. The solution is filtered by sixth-order implicit filters at the end of each time step.

Next to calculate the grid quality metric and the solution error (**??**), using the numerical and analytical solutions of the density at $t = 12$, the integrated values will be obtained for the whole domain (**??**, **??**).

$$E = \frac{(\rho - \rho_{exact})}{\rho_{exact}} \tag{2.7}$$

$$Q_I = \iint_S Q dx dy \tag{2.8}$$

$$E_I = \iint_S |E| dx dy \tag{2.9}$$

First of all, the problem will be solved for a uniform grid. Then, different grids will be tested to determine correlation between the grid quality metric and the solution accuracy.

- ***Uniform grid***

  This case is used as a baseline to calculate the solution error. The integrated values of the grid quality metric and the solution error are, respectively, $Q_I = O(10^{-9})$ and $E_I = O(10^{-5})$. This results verifies that the trucation error produced by a uniform grid is extremly small.

- ***Abrupt changes in the grid metrics***

  - ***Abrupt change in the grid spacing***

    In this study, an *abrupt change in the grid spacing* is obtained aplying the following expression to the grid coordinates of the uniform grid.

    $$x^*(\xi, \eta) = x(\xi, \eta) A_x H(x) \tag{2.10}$$

    (x, y) and (x^*, y^*) are the original and the modified grid coordinates, respectively. Also, A_x is a constant related to the grid spacing size and $H(x)$ is the Heaviside function.

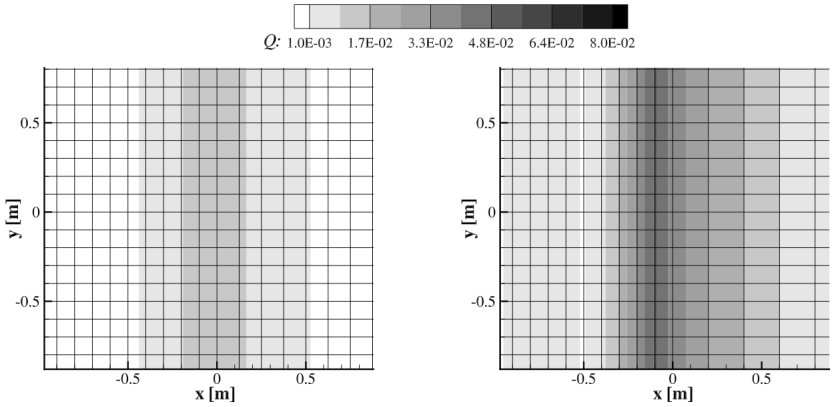


Figure 2.1: The effect on the grid quality metric when the grid spacing is changed at $x = 0$. Left: $A_x = 1.25$, right: $A_x = 2$

As we can see in the Figure **??**, more the grid spacing size increses, more the grid quality metric Q becames higher, which means a lower grid quality is obtained.

– ***Abrupt change in the grid line direction***

To get an *abrupt change in the grid direction* is imposed the next function to the uniform grid.

$$y^*(\xi, \eta) = y(\xi, \eta) + H(x)A_y x(\xi, \eta) \tag{2.11}$$

As in the equation **??**, (x, y) and (x^*, y^*) are the original and the modified grid coordinates and $H(x)$ is the Heaviside function. The constant A_y provides the grid line gradient.

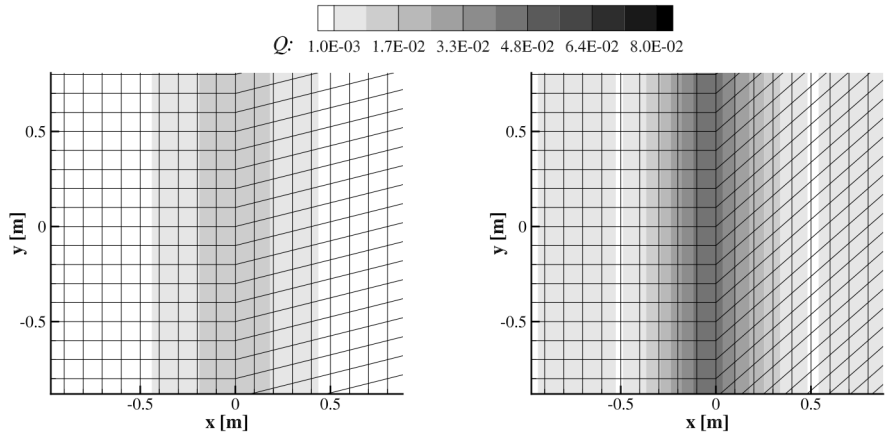


Figure 2.2: The effect on the grid quality metric when the grid line direction is changed at $x = 0$. Left: $A_y = 0.25$, right: $A_y = 0.875$

In the figure **??**, we can observe that the grid quality metric Q is increased as the grid line gradient A_y is higher. In conclusion, the solution accuracy is reduced.

- ***Skewed grid features***

  - ***Uniformly skewed grids***

    This case is an unusual geometric feature. However, *uniformly skewed grids* are obtained applying the following function to the grid coordinates of the uniform mesh.

$$x^*(\xi, \eta) = x(\xi, \eta) + A_s y(\xi, \eta) \tag{2.12}$$

    The original and the modified coordinates are (x, y) and (x^*, y^*), respectively. A_s means the grid line gradient across the entire domain. As we can see in the figure **??**, the solution error is not affected by the skew angle.

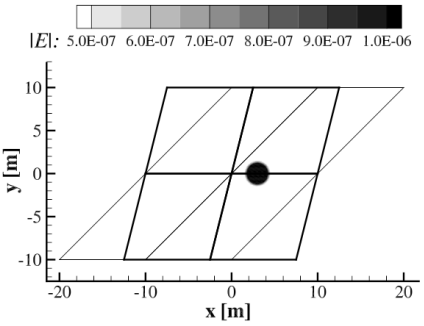


Figure 2.3: The effect on the solution error when the uniform cell skew is changed at $x = 0$. Thick black line: $A_s = 0.25$, thin black line: $A_s = 1$

  - ***Non-uniformly skewed grids***

    To get a *non-uniformly skewed grid*,it is applied the following trigonometric function to the uniform grid.

$$x^*(\xi, \eta) = x(\xi, \eta) + A_t \sin\left(\frac{2\pi y(\xi, \eta)}{L}\right)$$

$$y^*(\xi, \eta) = y(\xi, \eta) + A_t \sin\left(\frac{2\pi x(\xi, \eta)}{L}\right) \tag{2.13}$$

As it was written before, (x, y) and (x^*, y^*) are the original and the modified grid coordinates, respectively. A_t is the amplitude and L is the wavelength. In this case, the wavelength is set as $L = 10\,m$, while the amplitud A_t is changing.

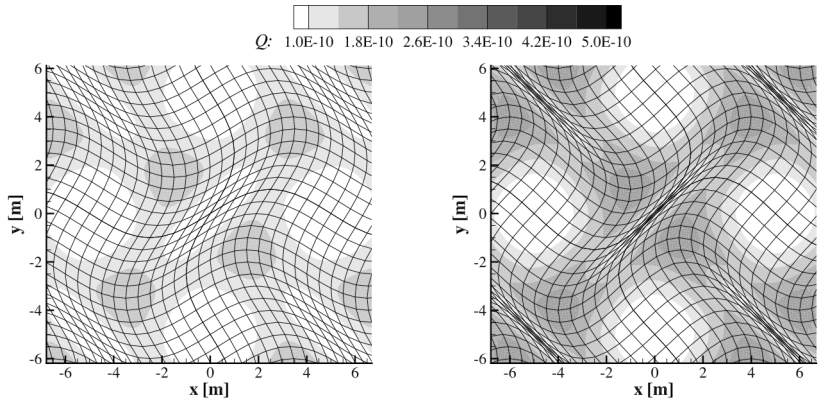


Figure 2.4: The effect on the grid quality metric when the amplitude of the non-uniformly skewed grids is changed at $x = 0$. Left: $A_t = 1$, right: $A_t = 1.5$

In the figure **??**, the largest value of the grid quality metric Q is focused on the areas where the variations on the grid direction are greatest. Then, in those positions the quality of the grid is lowest.

- ***Grid stretching***

  Grid stretching is normally used to increase the grid resolution close to a specific region. For this test case, the transverse space is kept uniform ($\triangle y = 0.1$), while the strem-wise space is defined by the following functions.

$$x^*(\xi) = A_h \tilde{x}(\xi)$$
$$x^*(\xi) = A_h \left[1 + \frac{\xi - max(\xi)}{\xi_{BL} - 1} (\tilde{x}(\xi_{BL} - \tilde{x}(\xi_{BL} - 1)) \right] \tag{2.14}$$

  A_h specifies the range where the grid are going to be stretched ($0 \leq x \leq A_h$). Outside this region, the stream-grid spacing is uniform. Also, ξ_{BL} is the number of points in the stretched region. The equations **??** are implemented when $1 \leq \xi \leq \xi_{BL}$ and $\xi > \xi_{BL}$. $\tilde{x}$ is the normalised grid point distribution function, defined as

$$s(\xi) = \frac{\xi}{\xi_{BL} - 1}$$
$$\tilde{x}(\xi) = 1 - \frac{\tanh(\alpha(1 - s(\xi)))}{\tanh(\alpha)} \tag{2.15}$$

  α is the parameter which determines the level of grid stretching and it is going to be changed in this test case. The other parameters are estabished $A_h = 1.5\, m$, $\xi_{BL} = 20$ and $max(\xi) = 101$. The result of increasing the grid stretching is a lower grid quality metric and, as we can see in the figure **??**, the numerical errors become higher.

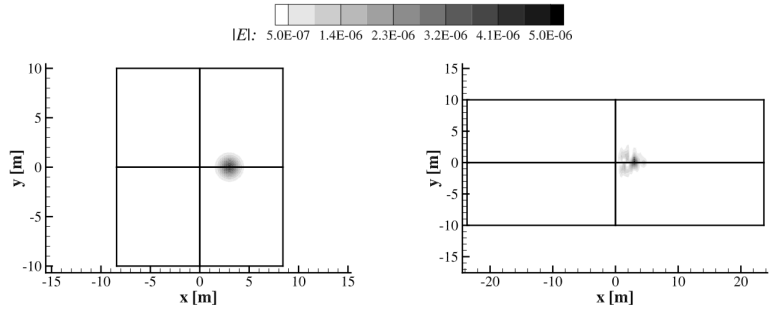


Figure 2.5: The effect on the solution accuracy when the grid stretching is changed at $x = 0$. Left: $\alpha = 0.5$, right: $\alpha = 3.5$

Finally, a correlation coefficient C_{QE} is defined to prove quantitatively the relationship of dependence between the integral grid quality metric, Q_I, and the solution error, E_I.

$$C_{QE} = \frac{\langle Q_I E_I \rangle}{[\langle Q_I^2 \rangle \langle E_I^2 \rangle]^{1/2}} \tag{2.16}$$

Grid Features	Explicit filter	Implicit filters
Grid spacing discontinuity	0.971	0.970
Grid direction discontinuity	0.955	0.954
Uniform skewness	0.427	0.128
Non-uniform skewness	0.921	0.921
Grid Stretching	0.932	0.913

Table 2.1: Correlation coefficient C_{QE}

In the table **??**, some correlation coefficients has been collected for the cases of study. We has omited two colums of results facilitated in the article (*Explicit filters w/CBC* and *Implicit filters w/CBC*) because only two out of five results are provided.

Also, as we can see in the results of the table **??**, grids with uniform skewness produce a trivial truncation error. However, the other kinds of grids have an important correlation coefficient and are most commonly used. The most significant error is generated by an abrupt change in the grid spacing.

In conclusion, the grid quality metric set out is strongly correlated to the solution error, provided that the grid induced error is the most important. This method allows a decrease of the computational cost and time required for the iterative process to get a great mesh. Moreover, it minimizes the use of additional filters to obtain a stable solution in high order solvers and allow us to know where the truncation error is more important, without having any knowledge of the solution field.

2.2 Large-eddy simulations of flow past a square cylinder using structured and unstructured grids

This article was written by *Yong Cao* and *Tetsuro Tamura* to study and discuss fundamentally the effects produced on the solution when *Large-Eddy Simulation, LES,* model is applied to a structured or unstructured mesh.

Firstly, the different types of grids are defined. Structured grids are identified as those that have regular connectivities between the elements and, in three dimensional space, are formed by hexahedra. In these kind of grids, convergence and accuracy are higher, but it is very difficult or impossible to use them in a complex geometry. However, unstructured grids are defined by an irregular connectivity, so their elements can be of arbitrary shape. Then, they are able to adapt to any geometric body.

Actually, in the present test case, a stationary square cylinder is past by a flow at $Re = 2.2 \times 10^4$ to examine the accuracy of large-eddy simulation. As a mather of fact, this simple geometry is used because it seems like high-rise buildings. Also, LES is usually used to simulate unstable flows, i.e. high Reynolds flows, which are of more interest in engeneering because of their complex flow development. There are many experimental data of this case, but the effects of free-stream turbulence and blockage ratio for the high Reynolds flow affect the aerodynamics characteristics and make difficult the comparison with numerical simulations that are not able to keep these experimental factors in mind. Nevertheless, as the computational power has increased in the last decades, it is possible to apply a fine mesh to LES. Consequently, the present study is going to investigate the effects of first and second order numerical schemes, to discuss how meshing strategies affect to the accuracy of prediction on unstructured grids and to examine the effect of the spanwise resolution and length.

Secondly, the numerical methods used are explained theoretically. Navier-Stokes and continuity equations for the incompressible LES can be written as

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j}, \quad \frac{\partial \bar{u}_j}{\partial x_j} = 0 \tag{2.17}$$

where the overline on the variables indicates a grid-scale filtering operation. τ_{ij} is known as the sub-grid scale (SGS) stress tensor and is defined by the following equation.

$$\tau_{ij} = \bar{u}_i \bar{u}_j - \overline{u_i u_j} \tag{2.18}$$

To model the stress tensor τ_{ij} can be used the dynamic Smagorinsky SGS model, in which the equation **??** is utilized.

$$\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} = 2\nu_e \bar{S_{ij}} = 2C\Delta^2 |\bar{S}| \bar{S}_{ij} \tag{2.19}$$

δ_{ij} is equal to unity if $i = j$ and zero in the other cases. $|\bar{S}|$ is the characteristic filtered rate of strain and is defined as $|\bar{S}| = (2\bar{S}_{ij}\bar{S}_{ij})^{1/2}$. In addition, Δ is the filter width and

S_{ij} is written as $S_{ij} = (\partial\bar{u}_i/\partial x_j + \partial\bar{u}_j/\partial x_i)/2$.

According to the type of mesh used with LES, it will proceed in a concrete way.

- *Structured LES*. The code is based on the Finite Difference Method (FDM) in a generalized curvilinear grid system. Then, the governing equations have to be transformed to the curvilinear coordinates because they are written in cartesian coordinates. The velocity and pressure values are accumulated in cell centers, while the contravariant velocity is stored in the cell faces. On the one hand, to obtain the solutions of velocity and pressure in time is used a fractional step method proposed by Kim and Moin (*Application of a fractional-step method to incompressible Navier-Stokes equations*). It consists of obtaining a first intermediate velocity, which is not correct. To correct it, it is used a Poisson equation. By last, using the correction, it is obtained the right velocity. This process is repeated until the last sub-time step and following that, the turbulent viscosity is computed using the dynamic Smagorinsky model. On the other hand, the Crack-Nicoloson scheme is applied to the viscous term and for the convective term, an explicit third order Runge-Kutta method is used. The convective term has been discretized approximating a fourth order central differencing scheme.

- *Unstructured LES*. It is utilized the open source toolbox known as OpenFOAM 2.3.0. It is based on Finite Volume Methods (FVM) and the flow is described in Cartesian coordinate system. Moreover, the numerical schemes applied are first and second order and the variables of flow are collected at cell centers. To solve in time the governing equations is used the PIMPLE method, which is based in the procedure of momentum predictor, pressure solver and momentum corrector. This method is a combination of SIMPLE and PISO methods. The pressure and viscous terms are discretized with a second order linear scheme. In this case, the convective term is linearized using different schemes: linearUpwind (second order upwind-biased scheme), linear up-wind stabilized transport (LUST, which is a combination of the central differencing scheme and the linear upwind schemes) and the limitedLinear (Total Variation Diminishing, TVD, scheme which is based on the Sweby limiter).

In conclusion, the main difference of numerical methods between structured and unstructured LES is found in the spatial discretization of the convective term, assuming the variancies in time schemes are not significant. Thus, in the present article is studied in more detail the convective term.

Thirdly, the computational cases are explained. The first LES simulation carried out was based on the structured grid using the FDM code and the computational domain size was $28D \times 24D \times 4D$.

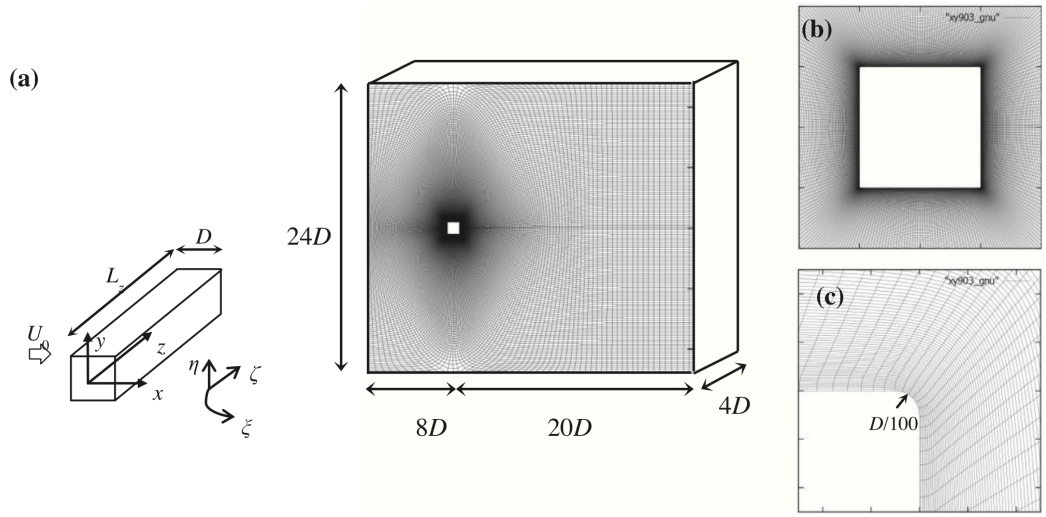


Figure 2.6: Computational domain size in a generalized coordinates. a) Total grid system; b) Near the square cylinder; c) Near the corner of the square cylinder with its radius of curvature

As shown the figure **??** (c), the corners of the square cylinder are rounded, where the radius of curvature is D/100. This small curvature does not affect the aerodynamic characteristics meaningfully. The boundary layer is solved accurately near the cylinder walls, setting the height of the cell as $\Delta y = 5.6 \times 10^{-4}$ and the growth ratio in the radial direction is 1.02. Also, the grids cells are spread out in the spanwise direction with an interval of 0.05D.

Then, the boundary conditions are settled down. The inlet boundary condition is a smooth and uniform inflow; the cylinder has a non-slip wall without any wall functions; the spanwise end boundary conditions are periodic; in the upper and lower boundaries, the free-slip boundary condition is used and in the outlet, the convective condition for all velocity components is set, as shown the equations **??** and **??**, respectively. The pressure on outlet is set as a convective condition, but for the other boundaries, it is stablished a zero-gradient condition.

$$\frac{\partial u_1}{\partial n} = 0, \quad u_2 = u_3 = 0 \tag{2.20}$$

$$\frac{\partial u_i}{\partial t} + U_c \frac{\partial u_i}{\partial x} = 0 \tag{2.21}$$

In the figure **??**, four cases with different grid resolution have been collected to compare the sensitivity of the numerical solution with the experimental data. The cell size has been reduced from Case 1 to Case 4 and there is no big difference between the pressure distributions. However, the grid resolution of the Case 2 has a better convergence to the experimental solution.

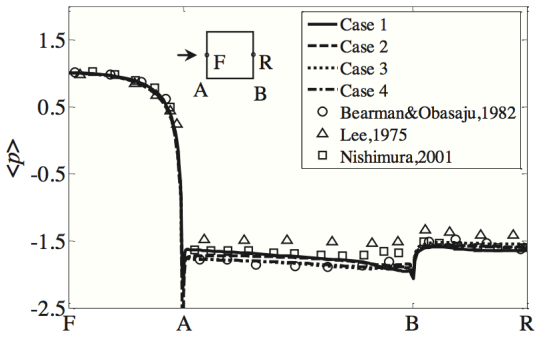


Figure 2.7: Time-average pressure distribution

In the present study, there are three main objectives to investigate: (1) effects of low-order numerical shemes; (2) meshing strategies; (3) effects of spanwise resolution and length. Then, different computational cases have been set up to be studied and the relationship between them is shown in the schematic figure **??**.

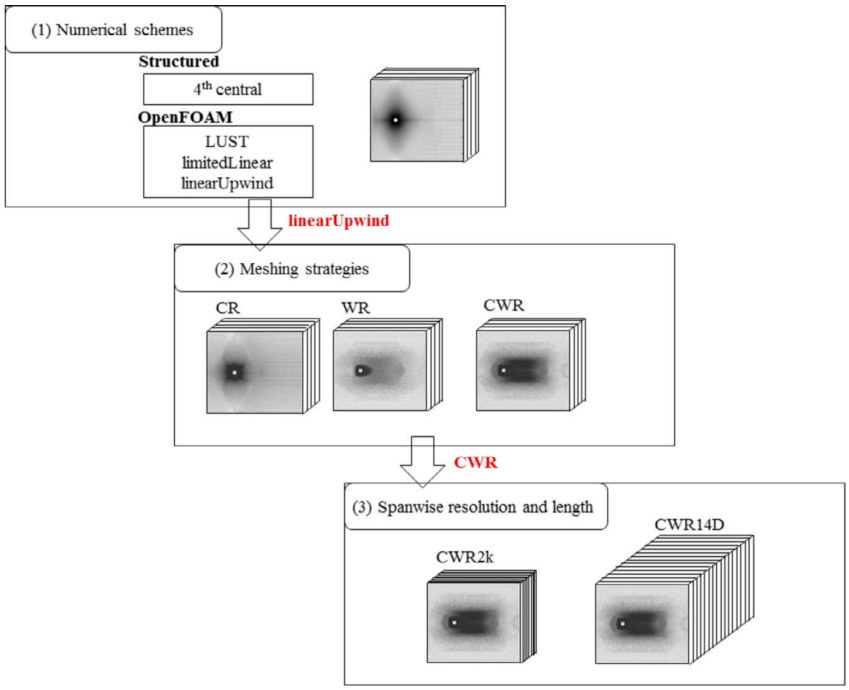


Figure 2.8: Schematic diagram for interpretation of relationship between computational cases

In the first group, four cases are studied, in which the grid topology is structured and the parameters are the same. The main differences are the codes and the numerical schemes used. The case named *structured* is calculated using the FDM code based on

the curvilinear coordinates and for the convective term, the numerical scheme used is fourth order central differencing. By contrast, the cases called *LUST*, *limitedLinear* and *linearUpwind* use the FVM code, in particular, OpenFOAM 2.3.0. For all the cases, the boundary conditions are the same, with the exception of the outlet in OpenFOAM, where the gradient of the velocity and the static pressure are defined as zero. After investigating the effects of numerical schemes, it has been concluded the second order upwind-biased scheme is more advisable compared with the others, so it will be adopted in the next groups.

The first and second order numerical schemes have been compared using the same grid, *structured*, in the first group. Thus, in the second group different meshing strategies are studied to improve the accuracy, although the numerical schemes are of low order. On the one hand, *CR* is a structured mesh which is refined near the cylinder. On the other hand, *WR* and *CWR* use hybrid grids to take advantage of the flexibility of the unstructured mesh. In concret, *WR* is a mesh in which the wake region is refined and *CWR* means cylinder and wake regions are refined. Finally, the *CWR* has been selected as the optimal mesh.

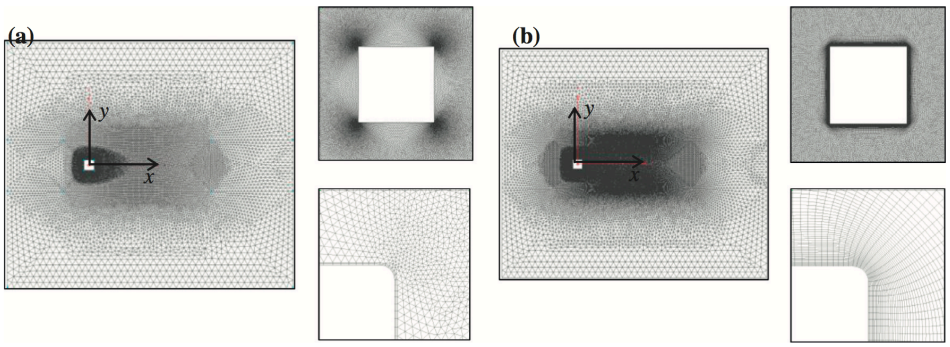


Figure 2.9: Hybrid grids. (a) Wake Refinament, WR; (b) Cylinder and Wake Refinement, CWR

In the third group, the effect of spanwise resolution and length is consedered, using the meshing strategy of CWR. Two cases are compared: *CWR2k*, in which the spanwise resolution is twice refined and *CWR14D*, in which the spanwise length is increased to 14D.

Additionally, the different cases computed have a CPU cost. In the first group, *linearUpwind* has the cheapest expense in CPU, followed by *limitedLinear* and *LUST*, which is 53% higher than *linearUpwind*. In the second group, *CWR* is $30\% - 50\%$ cheaper than *CR*. Finally, in the third group, *CWR2k* and *CWR14D* are 2.5 and 4.4 times as expensive as *CWR*.

Fourthly, the results obtained have been compared and discussed. To carry it out, this

part in the article was divided into five subparts, in which different aspects are analysed and commented on.

- ***Comparison of global quantities***

  To select the experimental data, it is important keeping in mind that the experimental conditions as the aspect ratio of the cylinder (AR), the free stream turbulence intensity (I_u) and the blockage ratio (BR) can affect the results. Thus, it was defined the $AR > 9$, $I_u < 1\%$ and $BR \sim 5\%$ to choose the experimental data.

  Firstly, in terms of effects of numerical schemes, the case called *Structured* predicts accurately the aerodynamic responses in comparison with the experiments. However, the cases using OpenFOAM (*LUST*, *limitedLinear* and *linearUpwind*) estimate the time-averaged values ($\bar{C}_D$), the Strouhal number (St) and the formation length (L_f) consistently with experiments, but the fluctuating drag ($C_D^{'}$) and lift coefficients ($C_L^{'}$) are larger than experiments.

  Secondly, the results of different meshing strategies (*CR*, *CW* and *CWR*) are studied. The time-averaged drag and the Strouhal number are quantitatively close to the experimental data. However, the fluctuating lift coefficient is still higher than experiments.

  Thirdly, the aerodynamic characteristics remain constant in the case of *CWR2k*. Nevertheless, when the spanwise length is increased, *CWR14D*, the fluctuation lift and drag coefficients experiment an important reduction.

- ***Comparison of pressure distributions***

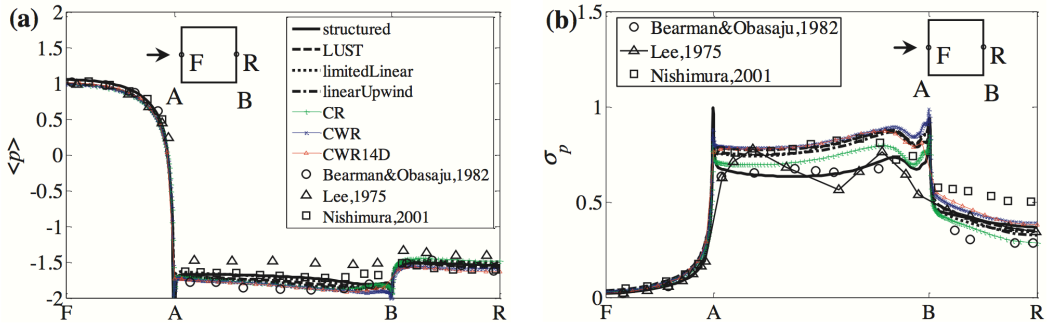


Figure 2.10: Distribution of pressure on the cylinder surface: (a) time-averaged pressure; (b) fluctuating pressure

On the one hand, figure **??** (a) shows the distribution of time-averaged pressure. As we can see, all cases tested in the present study have similar results to experiments.

It means that the low-order numerical schemes and meshing strategies used are sufficient to reproduce the time-averaged quantities. On the other hand, figure **??** (b) represents the distribution of r.m.s values of pressure coefficients. In general, the numerical results obtained are qualitatively rational compared with experiments, but the values of pressure on the side faces are quite different. If we observe the numerical results, we will see that the fluctuating pressure obtained by the *structured* case is lower than the results obtained with OpenFOAM cases.

- ***Comparison of wake flow***

  The velocity profiles along the wake center line are shown in the figure **??**, where are included the experiments, the previous numerical simulation and the actual cases studied.

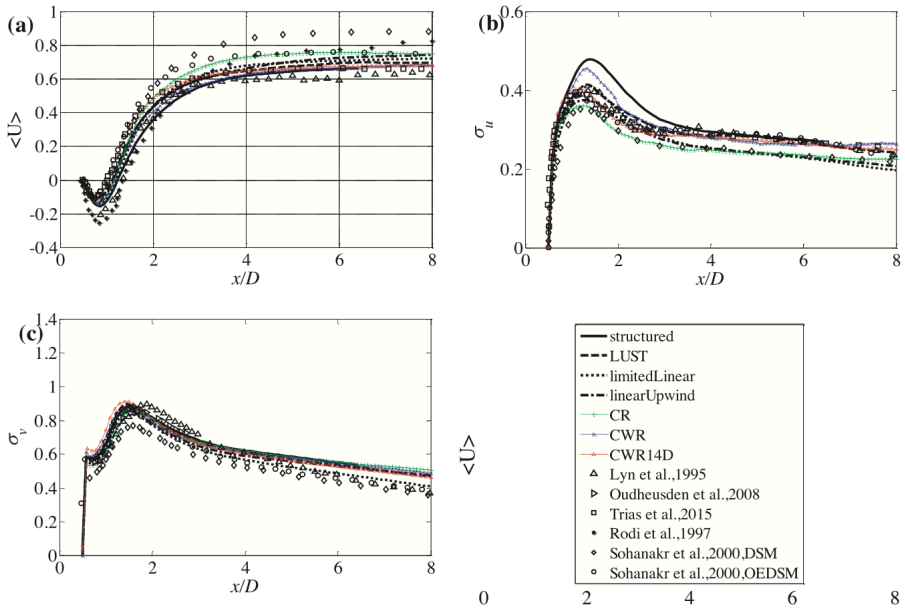


Figure 2.11: (a) Time-averaged stream-wise velocity; (b) Fluctuation of stream-wise velocity; (c) Fluctuation of the vertical velocity

In the figure **??** (a), we can see in the wake region ($x/D > 3$) that the LES studies of Rodi et al. and Sohankar et al. tend to over-predict the time-averaged velocity, while the other numerical cases obtain similar results. In fact, *structured*, *CWR* and *CWR14D* cases have a better agreement with the emperiments and DNS results. Near the wake region ($x/D < 2$), all the cases of the present study overestimate the time-averaged stream-wise velocity and, in the intermediate wake ($2 < x/D < 3$), they obtain lower values than Lyn et al. results.

Now, figure **??** (b) is going to be explained in detail. In the wake region ($x/D > 3$), the cases called *structured*, *CWR* and *CWR14D* have results close to those of Lyn et al. However, the cases without a wake refinement, i.e. *limitedLinear*, *linearUpwind*

and *CR*, underestimate the fluctuation of stream-wise velocity. Moreover, in the near-wake region ($x/D < 2$), the *structured* and *CWR* cases obtain larger values than the others. In the case of *CWR*, it is greater than *CWR14D* because of the extension of the spanwise length.

Finally, figure **??** (c) shows the fluctuating cross-stream velocity along the center-line. In general, all the numerical cases tested in this study are consistent with the experimental results. In the far wake, the results are a bit larger than the precedent numerical simulations. This could be the consequence of having a finer mesh. Also, it is important to highlight that the fluctuating cross-stream velocity of *limitedLinear* is smaller, probably on account of TVD scheme.

In conclusion, the numerical schemes tested in the present study approximate accurately the results in wake prediction, using the current grid system and cost. In order to obtain more exact results with *CWR* using similar grid resources, the grid should be refined with special care.

- ***Comparison of shear layer flow***

  To predict accurately the shear layer is an important aspect to obtain rigurous results. In general, the topology of the streamlines is very similar in the numerical cases. In the figure **??**, two flow patterns of time-averaged velocity are represented for *Structured* and *CWR* cases.

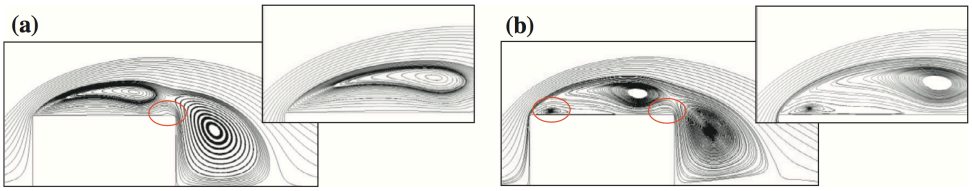


Figure 2.12: Streamlines of time-averaged velocity in the cases: (a) Structured; (b) CWR

As we can see in the figure **??**, there are some differences between both cases. In the figure **??** (b), two recirculation regions called secondary vortex are marked with a red ellipse. By contrast, the frontal secondary vortex does not appear in the *Structured* case. As a result of these vortex, the velocity near the wall experiencies a sign change.

In figure **??**, the wall shear stresses are plotted for all cases. F and R are the frontal and rear stagnation points, whereas C and D are the frontal and leeward corners of the upper surface. These graphics are important because when they show a nevative shear stress means that the flow is travelling in the inverse direction. As we can see in the figure **??** (c), the region with negative shear stresses coincide with the leeward secondary vortex mentioned before. Among the numerical results, the

structured and *CWR* have smaller absolut values of the negative shear stresses, but in general, all results are approximated to the DNS study.

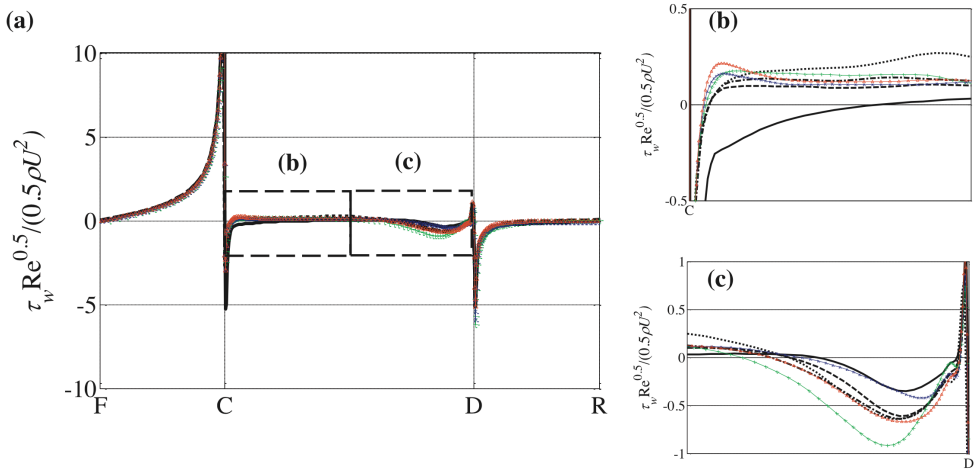


Figure 2.13: Distributions of wall shear stresses. (a) From F to R; (b) from C to the middle of the upper cylinder wall; (c) from the middle of the upper cylinder wall to D

Additionally, the profiles of velocity along the vertical direction have been plotted in figure **??**. The measures are performed in the position indicated as b. Some quantitative differences exist between the numerical cases. In the *structured* case are obtained results of time-averaged and fluctuating cross-stream velocity lower in the region near the shear-layer. The case of *CR* has the biggest results because in this case, most of the cells are around the cylinder, so the prediction of the shear layer is more accurate.

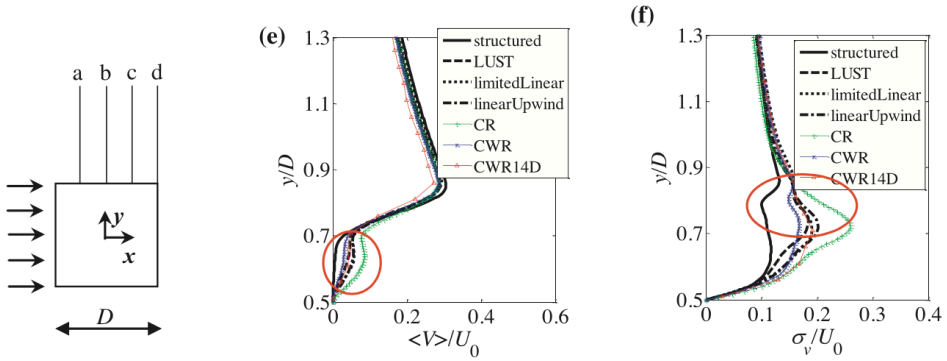


Figure 2.14: Velocity profiles along the vertical direction in the shear-layer region. (e) Time-averaged cross-stream velocity when $x/D = 0$; (f) fluctuating cross-stream velocity when $x/D = 0$

It is considered that a large fluctuation cross-stream velocity is related with the roll

up of the shear layer about the side faces. Thus, to prove it, the roll-up of the shear layer is shown in the figure **??** for the *linearUpwind* and *structured* cases, where the fluctuation of the *linearUpwind* case is greater.

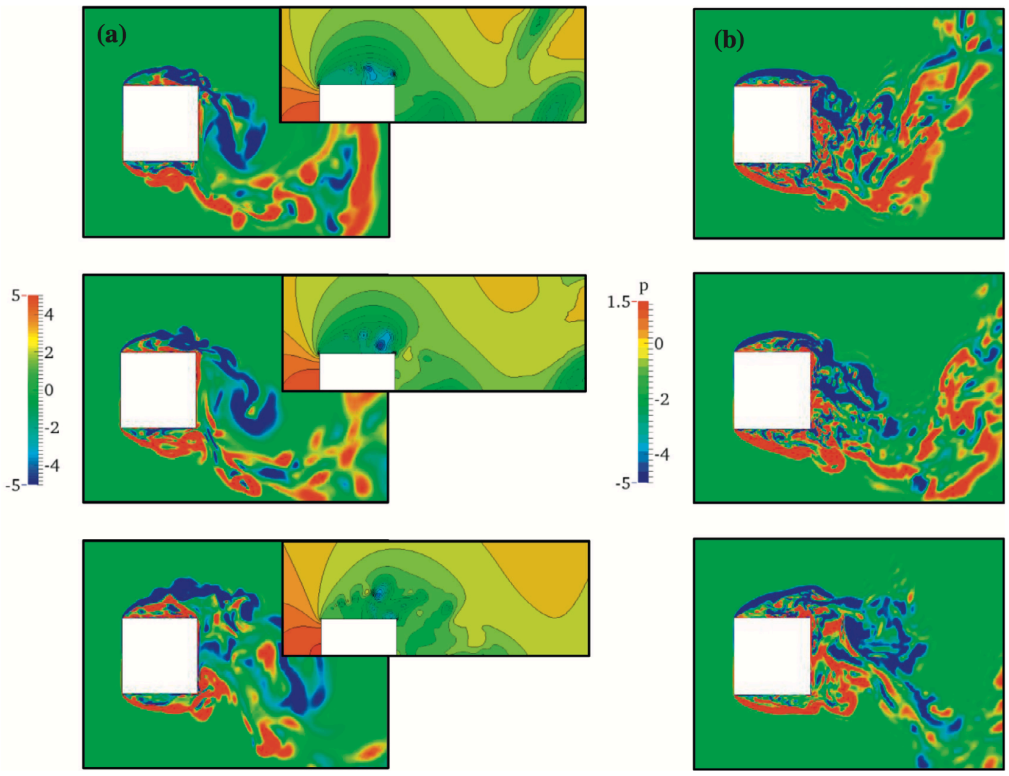


Figure 2.15: Time evolution of instantaneous spanwise vorticity and pressure for (a) linearUpwind; (b) structured

If we focus our attention in the upper side face of the cylinder, in the case of *linearUpwind* we can see the shear-layer vortices after the separation from the frontal corners. On contrary, in the *structured* case, the shear layer is smooth and does not show the secondary vortex in the frontal corner, possibly leading to a smaller fluctuating cross-stream velocity. Then, there is a consistent relationship between the results obtained for the cross-stream velocity (figure **??**) and the wall shear stresses (figure **??**). It can be concluded that the wall shear stresses are responsible for the desviation of the cross-stream velocity.

Finally, it is known that Kelvin-Helmholtz (K-H) inestabilities appear on the separated shear layers at high Reynolds numbers. OpenFOAM is able to predict these instabilities, which do not generally affect the forces acting on the cylinder.

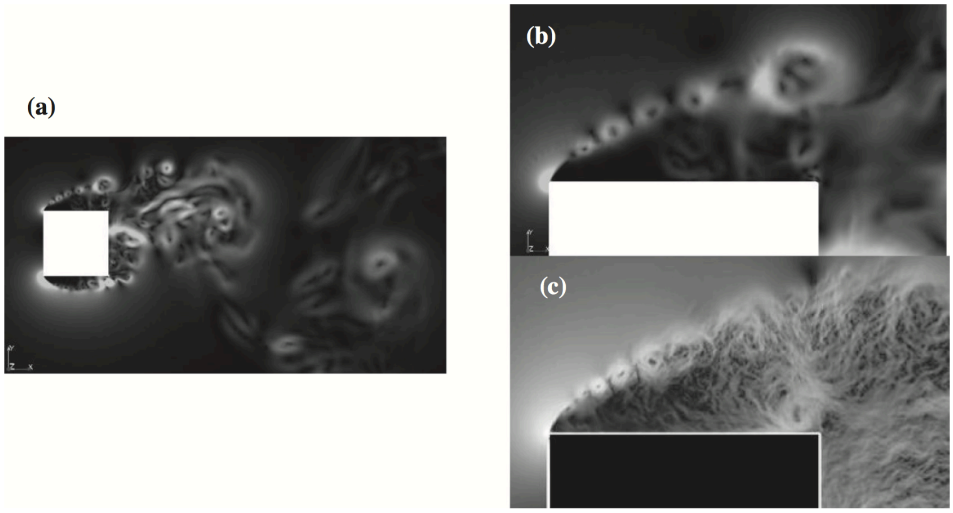


Figure 2.16: Visualization of Kelvin-Helmholtz structures. (a)-(b) CWR case; (c) DNS by Trias et al.

As we can see in the figure **??**, OpenFOAM provides a satisfactory simulation of the K-H structures with respect to the DNS results. However, compering the figures **??** (b) and (c), it seems OpenFOAM provides a lower prediction of the wavenumber and the first K-H structure is more downstream than in the DNS simulation.

- ***Discussion on effects of spanwise length***

The selection of the spanwise length is a significant choice because of the cell number and the computational cost. As we said before, the r.m.s values of pressure are similar in the cases of *CWR* and *CWR14D*, the fluctuation of total lift is decreased about 20% when the spanwise length is increased to 14D.

Then, a spanwise correlation coefficient of pressure has been calculated using the following equation

$$R_{pp}(s) = \frac{\langle \Delta p'(z_0) \Delta p'(z_0 - s) \rangle}{\sigma_{\Delta p'}(z_0) \cdot \sigma_{\Delta p'}(z_0 - s)} \tag{2.22}$$

where s means the spanwise distance and $\Delta p'$ is the difference of pressure mesured at the middle of two opposing side faces. The results are plotted with the experimental results of Vickery in the figure **??** (a). As we can see, the spanwise correlation of *CWR14D* has a similar performance than the experimental results. In the figure **??** (b), the difference of pressure at the middle of two opposing side faces is plotted (ΔC_p). Also, the sectional lift and drag are calculated and represented in the figure, with the pressures at the middle of side faces ($C_{p,upper}$ and $C_{p,bottom}$). The lift coefficient decreases when the spainwise length is increased to 14D.

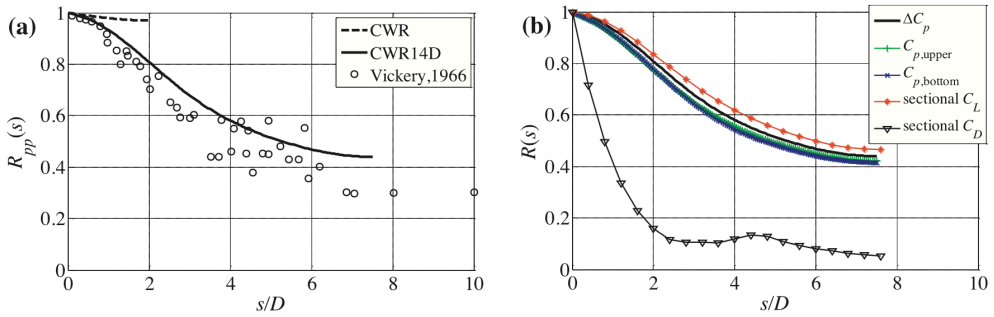


Figure 2.17: (a) Spanwise correlation coefficient of pressure; (b) Spanwise correlations based on different pressure quantities

Lastly, it is shown in the figure **??** a flow visualization at the dimensionless instant $t^* = 178.7$. We can see from figures **??** (a) and (b) the phase difference of vortex shedding which takes place along the span. Concretely, figure **??** (b) shows the distribution of spanwise vorticity at two cross sections, where the phase difference is about π. Also, in the figure **??** (c), it is possible observe the phase difference induced in the pressure along the span. This phase difference of pressure affects to the reduction of the spanwise correlation when the spanwise length is increased to 14D, as we see in the figure **??**.

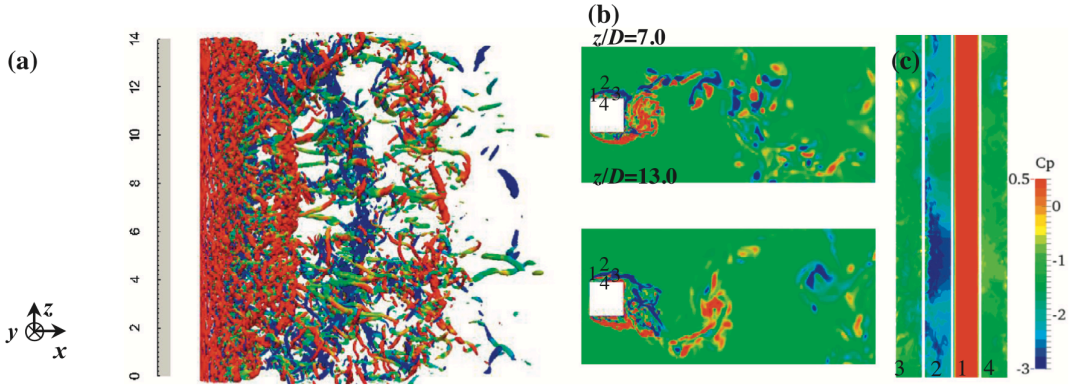


Figure 2.18: (a) Instantaneous iso-surfaces at the dimensionless instant $t^* = 178.7$; (b) Distributions of spanwise vorticity components on the cross-sections of $z/D = 7.0$ and $z/D = 13.0$; (c) Pressure distributions on four cylinder surfaces

Last but not least, after examinating the accuracy of LES on structured and unstructured grids, the main conclusions reached are the following:

- The low order schemes used by OpenFOAM do not influence the accuracy of prediction.
- The typical schemes of OpenFOAM give us a similar prediction of time-averaged and r.m.s quantities, but the *limitedLinear* provides an unnatural distribution in the inertial subrange of turbulence motions.

- The refinement of meshing in the wake region (*CWR*) improve the results of the time-averaged and fluctuating velocity.
- The prediction of the flow characteristics depends on the scheme used and influences the velocity profiles over the shear layer region and the fluctuation of streamwise velocity behind the cylinder. The case of *CR* with refinement of hexahedra cells gives a similar solution to previous DNS results.
- The Kelvin-Helmholtz inestability can be accurately predicted by OpenFOAM.
- The increase in spanwise resolution does not affect the prediction, but the increase of spanwise length induces a phase difference in the vortices along the span, reducing the fluctuation of lift.

Chapter 3

CFD Methodology

Computational Fluid Dynamics, CFD, is the science that studies the numerical simulation of fluid flows. It is a useful tool based in the Fluid Mechanics equations that gives us the oportunity to recreate experiments and obtain quantitative predictions in an accurate way.

The goal of this thesis is to study and verify that the quality of a mesh is related to the accuracy of the solution. So, we are going to simulate a flow with high Reynolds number around a squared cylinder using different types of meshes to discretize the domain. In fact, this kind of issue is of great interest, specially in the field of construction, due to it is very important to obtain precise results, as it has been said before.

There are many CFD codes, but we are going to use *ANSYS: Fluent/CFX* to complete this Thesis. Generally, in CFD simulations, the basic scheme could be defined as:

- Preprocessing
- Solver
- Postprocessing

Each part is composed of different elements, which are explained in detail below.

3.1 Preprocessing

Firstly, *Preprocessing* is carried out. It involves the definition of: the *concept of the problem*, the *geometry*, the *mesh*, the *model and flow attributes* and the *boundary conditions*.

3.1.1 Concept of the problem

Before starting to work, the main aspect is to define an aim of study. Then, we have to adapt the real situation to a feasible problem, without losing the consistence (i.e. dimensions, symmetry, stationary/transitory).

Once the problem has been established, it is possible to search bibliography related with the case and gather empirical results. In this way, we could estimate the numerical results we should obtain and how much time will be required to complete the project.

3.1.2 Geometry

The next step is to define the geometry. There is the possibility of creating a simple geometry using a tool of the simulation program (*ANSYS Workbench*) or to import a realistic geometry done with a CAD program. It is important to define correctly the calculation domain to obtain reasonable results.

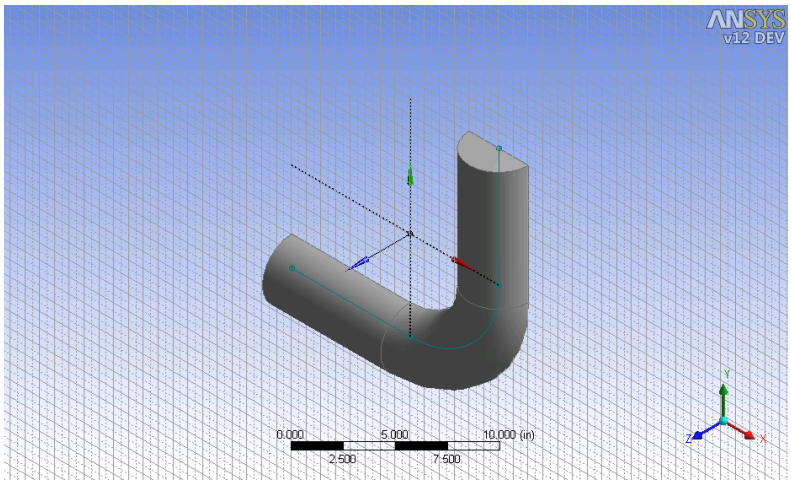


Figure 3.1: Pipe Geometry

3.1.3 Mesh

To generate the *Mesh* means to discretize the spatial domain in small elements connected between them, in order to calculate the flow variables using numerical methods. It is one of the most important aspects in the CFD simulation because it is related to the computational cost, the accuracy and convergence of the solution. In essence, there are three types of meshes (Figure **??**):

- *Structured mesh*

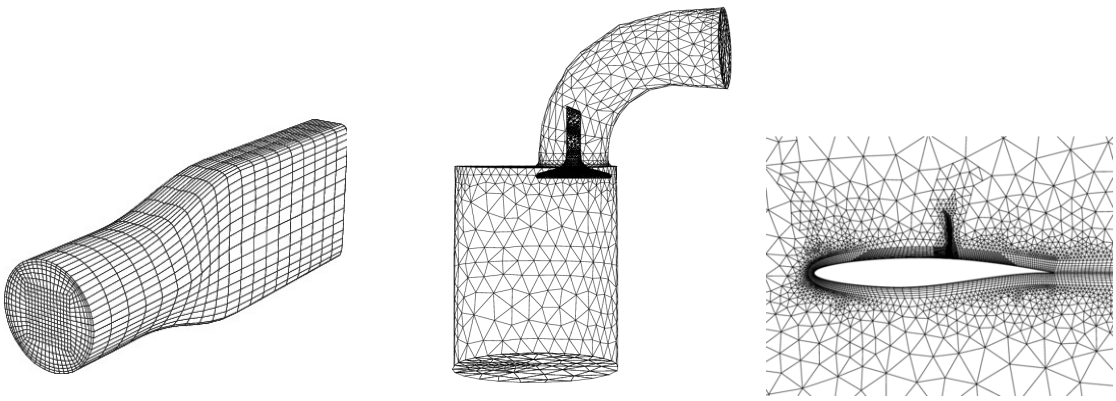
  There is a regular connectivity between the elements and it is composed by regular elements like quadrilateral in 2D and hexahedra in 3D. As the space is divided with such efficiency, each point is defined by indexes (i, j, k) in the cartesian coordinate system. Also, this type of mesh can be classified as *orthogonal* or *non orthogonal*. A structural mesh will be orthogonal if the lines intersect perpendicularly.
  Some advantages could be the ease of access to the information stored in each node and the high accuracy of the solution if the mesh is lined up with the streamlines. A disadvantage is the difficult implementation in complex geometries.

- *Unstructured mesh*

  This kind of mesh is identified by irregular connectivities between the elements and is generated by an algorithm of the program. For this reason, the nodes do not follow a particular order, so they cannot be indentified by indexes. In 2D, the typical elements used are a mix of quadrilaterals and triangles and, in 3D, tetrahedra and hexahedrons.
  The principal advantages of unstructured meshes are that they can be used with complicated geometries and reduced in size near the body. Nevertheless, the disadvantages would be the large space it takes up in the memory of the computer and how easy is it to get an unsuitable mesh.

- *Hybrid mesh.*

  In this mesh, structured and unstructured meshes are combinated efficiently. Structured grids can be used in the parts where the geometry is regular, while unstructured meshes are used in those that are complex.



(a) Structured (b) Unstructured triangular (c) Hybrid

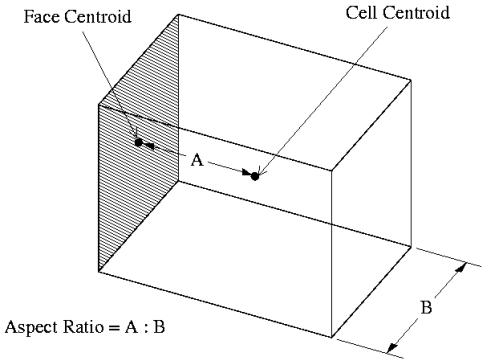
Figure 3.2: Types of grids

The main objective of a mesh is to capture all the flow features. To achieve it, the mesh near the wall should be fine enough to resolve the boundary layer, increasing the size of the cells away from it. It should be note that must exist a balance between the size of the cells and the computational cost of the simulation. In addition, another important aspect of the mesh is the grid quality. The grid quality is related to the accuracy and the stability of the solution. The main attributes associated with the suitability of the mesh are the *aspect ratio*, *orthogonality* and *skewness*.

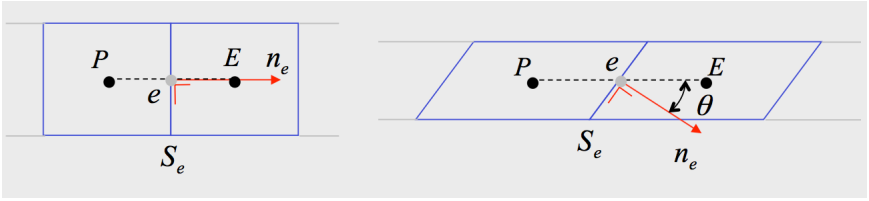
- *Aspect Ratio, AR.* The aspect ratio is known as a measure of the stretching because it is defined as the ratio of the maximum distance between the cell and face centroids to the minimum distance between the nodes of the cell. Ideally, it should be equal to 1 to ensure the best results. Moreover, the local variation in cell size should be minimal, it means the adjacent cell should not have an increase of aspect ratio higher than 20%.

- *Orthogonality.* It is referred to the angle, θ, between the line which connects the nodes of adjacent cells and the normal vector of the face dividing these relative cells. In 1D, the orthogonality of the grid is always verified but in 2D, it could not be. To obtain good results, this angle θ should approach to zero.

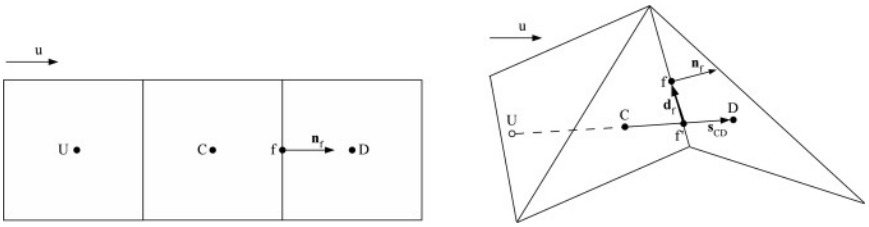
- *Skewness.* The skewness is defined as the distance **d** between the line which joins two adjacent nodes and the point in the middle of the face intersecting these cells. The error increases when this distance is far from zero.



(a) Aspect ratio



(b) Orthogonality

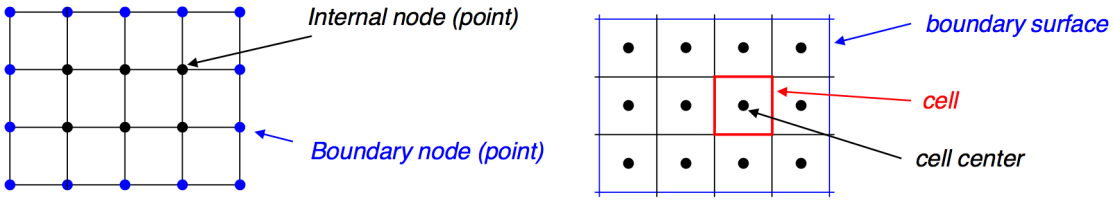


(c) Skewness

Figure 3.3: Grid quality

The numerical methods used to discretize the equations define the place where the information about the flow variables is collected (*nodes* or *cell centers*). In general, the numerical methods are used to discretize the partial derivates or the integrals, depending on the case, in the governing equations using the variables at the grid points. In other words, it means to approximate the governing equations by a system of algebraic

equations in which the variables are calculated in a finite number of locations in space and time. Actually, in a CFD study, the governing equations which we want to solve are the equations of mass, momentum and energy balance. This governing flow equations are non-linear so, to resolve them, we should linearize those equations using a numerical method. Three numerical methods stand out: *Finite Differences*, *Finite Volumes* and *Finite Elements*. These numerical methods are going to be detailed in the section **??**. The main effects produced when we use *Finite Differences Methods* are the governing equations are written in differential form and the flow variables are obtained for each node (Figure **??** (a)). By contrast, using *Finite Volumes Methods* the governing equations will be written in integral form and the flow variables are calculated in the cell center (Figure **??** (b)).



(a) Diferential equations (b) Integral equations

Figure 3.4: Effects produced on the mesh by numerical methods

In conclusion, the discretization of the space using a mesh allows us an approximate solution, but it could contain some errors. On the one hand, the solution will be affected by the truncation errors of the numerical methods, which can be decreased using models of higher order. On the other hand, the low quality of the mesh is associated with an error in the solution. We can solve this kind of error increasing the accuracy of the grid.

3.1.4 Model and flow attributes

The behaviour of a fluid is defined by a mathematical models based on the hypothesis in which is said the matter is continuous. In this way, each point of the flow field has associated features as density, velocity, pressure and temperature, among others. This hypothesis is correct in the equlibrium thermodynamics, which is fullfilled in the flows of interest.

In particular, the *velocity* is a characteristic used to classify the types of flows. On the one hand, if particles of a fluid move at low speeds, they will describe a regular path. This kind of flow is called *laminar*. On the other hand, at high speeds, the movement becames unsteady and is known as *turbulent* flow. Also, the *Mach number* is used to determine the type of fluid. The *Mach number* is defined as the ratio between the flow speed and the sound speed. For low Mach numbers, as $Ma < 0.3$, the flow is considered *incompressible*, which means the density is approximately constant. However, at higher Mach numbers, the flow is *compressible*. All these features of the flow determine the form

of the governing equations and the types of numerical methods used to solve them.

Generally, the motion of a viscous fluid is described by the governing equations known as *continuity*, *momentum* and *energy equations*. These balance equations arise from applying Newton's second law ($F = m \cdot a$) and from assuming that mass and energy are conserved. They can be expressed in different forms according to the type of control volume which is taken, but in all cases they mean the same.

- **Mass balance equation**

  It is obtained applying the phisical principle of the mass conservation to a control volume. If we consider a finite and fixed in space control volume, we will obtain the integral conservative form:

  $$\frac{\partial}{\partial t}\int_V \rho dV + \int_S \rho \mathbf{v}\cdot\mathbf{n}dS = 0 \tag{3.1}$$

  Using the Gauss Theorem,

  $$\int_S \rho \mathbf{v}\cdot\mathbf{n}dS = \int_V \nabla\cdot(\rho\mathbf{v})dV$$

  and assuming the equation **??** is valid for any volume, the result is the conservative differential form.

  $$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\mathbf{v}) = 0 \tag{3.2}$$

- **Momentum balance equation**

  This equation is based on the second Newton's law. The integral conservative form is

  $$\frac{\partial}{\partial t}\int_V \rho\mathbf{v}dV + \int_S \rho\mathbf{v}\mathbf{v}\cdot\mathbf{n}dS + \int_S p\bar{\mathbf{I}}\cdot\mathbf{n}dS - \int_S \bar{\tau}\cdot\mathbf{n}dS = \int_V \rho\mathbf{f}dV \tag{3.3}$$

  where $\mathbf{f}$ are the forces per unit mass and $\bar{\tau}$ the shear stresses. In a Newtonian fluid, the shear stresses are proportional to the velocity gradients.

  $$\tau_{ij} = \delta_{ij}\lambda\nabla\cdot\mathbf{v} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

  Being $\delta_{ij} = 0$ if $i \neq j$ and $\delta_{ij} = 1$ when $i = j$.

  Also, we can write the momentum balance equation in a differential and conservative form for a infinitessimal and fixed control volume.

  $$\frac{\partial(\rho\mathbf{v})}{\partial t} + \nabla\cdot(\rho\mathbf{v}\mathbf{v}) + \nabla p - \nabla\cdot\bar{\tau} = \rho\mathbf{f} \tag{3.4}$$

  The momentum equations for a viscous flow are called *Navier-Stokes equations*. Also, for an inviscid flow, they are known as *Euler equations*.

- **Energy balance equation**

  Last but not least, the energy balance equation is based on the physical principle that affirms *energy is conserved*, which is expressed by the First Principle of Thermodynamics. The differential conservative form of the balance equation is

$$\frac{\partial E}{\partial t} + \nabla \cdot [(E+p)\mathbf{v}] - \nabla \cdot (\bar{\tau} \cdot \mathbf{v}) + \nabla \dot{\mathbf{q}} = \rho \dot{\xi} + \rho \mathbf{f} \cdot \mathbf{v} \tag{3.5}$$

  where $\dot{\xi}$ is the heat absorbed per unit time and unit mass and E is called *total energy per unit volume* and is defined as $E = \rho\left(e + \frac{1}{2}|\mathbf{v}|^2\right)$. Furthermore, if we integrate the equation **??** over a finite volume and use the Gauss Theorem, we obtain the integral conservative form.

$$\int_V \frac{\partial E}{\partial t} dV + \int_S (E+p)\mathbf{v} \cdot \mathbf{n} dS - \int_S (\bar{\tau} \cdot \mathbf{v}) \cdot \mathbf{n} dS + \int_S \dot{\mathbf{q}} \cdot \mathbf{n} dS = \int_V \rho \dot{\xi} dV - \int_V \rho \mathbf{f} \cdot \mathbf{v} dV \tag{3.6}$$

At the end, we have five governing equations: one equation for the conservation of mass, three for the momentum balance and one for the conservation of energy. Nevertheless, the unknown variables are six: ρ, $\{u, v, w\}$, p, e. So, one equation more is needed to close the system. This equation is called *constitutive law* because it describes the characteristics of the fluid or the thermodynamic transfomation that takes place. For example, in the typical aerodynamics applications, the gas is assumed as a perfect gas. Then, the sixth equation is the *perfect gas law*:

$$p = \rho R^* T$$

in which the temperature T is present, instead of the internal energy e. However, once two state variable are known, a third can be obtained as a function of them ($e = f(p, T)$).

Presently, most of the flows of interest in engineering are turbulent. *Turbulence* is a flow regime characterized by chaotic variations of its propierties and composed by eddies of different sizes. The principal properties of turbulent flows are the following:

- *Highly unsteady*. A plot of the velocity as a function of time could seem random.
- *Tri-dimensionality*. The quantities of interest, as time-averaged velocity, can be considered as a function of two coordinates, but the instantaneous velocity field fluctuates rapidly in all three spatial dimensions.
- *Great deal of vorticity*. In fact, the intensity of turbulence is increased by vortex stretching mechanism.
- *Turbulent diffusion*. In this process, the fluid parcels with different concentrations of at least one of the conserved properties are mixed by diffusion.
- *Constant growth of internal energy*. Due to the process just mentioned, turbulence puts fluids of differing momentum content in contact. The kinematic energy of the flow is reduced by the action of the viscosity, which is a dissipative process. Then, the lost energy is converted into internal energy of the fluid.

- *Difficult study*. In turbulent flows, there are coherent structures, i.e. repeatable and deterministic events. Nevertheless, the random component of turbulence causes that these events differ from one to each other in size, strength and time interval between occurences.

- *Fluctuation on a broad range of length and time scales.*

As a matter of fact, when the turbulence flow is observed, different vortex structures, which are called eddies, are noticed. There are eddies of different sizes and inside them, significant velocity variations take place. Due to the presence of different scales in the flow, various Reynolds numbers can be defined as

$$Re_\lambda = \frac{\lambda v_\lambda}{\nu} \tag{3.7}$$

where λ is the dimension of the spatial scale. When λ is large, Re_λ is greater and the vicous diffusion is insignificant. Nonetheless, at the smallest scales, $Re_\lambda \sim 1$, the viscosity becames important. These scales are known as *Kolmogorov scales* and the spatial dimension is represented with η. The intermediate scales are known as *inertial scales* because they transport the energy from largest to smallest scales. The process which takes place is the following: turbulent kinetic energy goes from big eddies to small ones, until it is dissipated into heat by the viscosity, as we can see in the figure **??**. This process is known as *Turbulent energy cascade*.

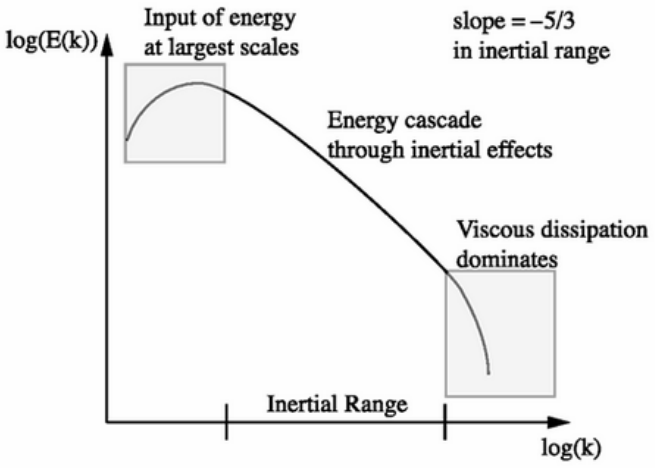


Figure 3.5: Energy cascade in turbulent flows

Turbulent flow is described by governing equations similar to those of laminar flow, but their resolution is much more difficult and expensive. It is because turbulent flows contain variations on a wider range of length and time scales than laminar flows. There are different numerical methods to compute these types of flows.

- *Reynolds-Averaged Navier-Stokes, RANS*

  In this method, all scales of turbulence are modeled. First of all, in a turbulent flow, every variable ϕ can be approached statistically as a mean component $\overline{\phi}$ and a fluctuating component ϕ':

$$\phi(x_i, t) = \overline{\phi}(x_i, y) + \phi'(x_i, t). \tag{3.8}$$

  where the mean component in an steady flow can be written as

$$\overline{\phi}(x_i) = \lim_{T\to\infty} \frac{1}{T} \int_0^T \phi(x_i, t)dt \tag{3.9}$$

  where T is the averaging interval, which must be large than the typical time scale of the fluctuations. However, if the flow is unsteady, we will use

$$\overline{\phi}(x_i, t) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \phi(x_i, t) \tag{3.10}$$

  where N is the number of members in the sum and must be large enough to eliminate the effects of the fluctuations.

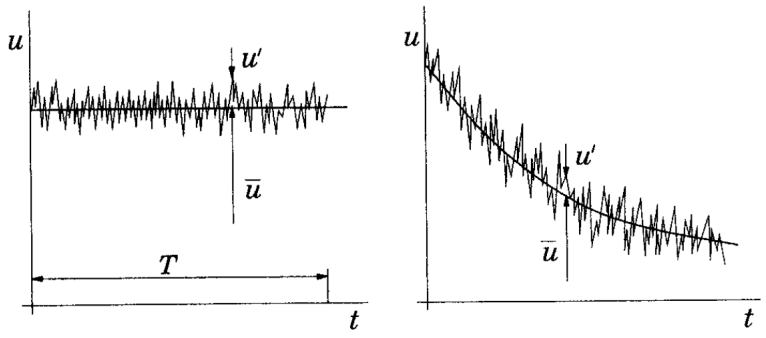


Figure 3.6: Steady and unsteady flows, respectively

Applying one of these averaging processes, called *Reynolds averaging*, to the *Navier-Stokes equations*, we obtain the *Reynolds-averaged Navier-Stokes equations*, in this case written for incomplessible flows and in Cartesian coordinates as

$$\frac{\partial(\rho\overline{u}_i)}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho\overline{u}_i\overline{u}_j + \rho\overline{u_i{}'u_j{}'}\right) = -\frac{\partial\overline{p}}{\partial x_i} + \frac{\partial\overline{\tau}_{ij}}{\partial x_j} \tag{3.11}$$

where $R_{ij} = \overline{u_i{}'u_j{}'}$ is called *Reynolds-stress tensor*, $\overline{u}_i\overline{u}_j$ is known as the *turbulent scalar flux* and $\overline{\tau}_{ij}$ are the mean viscous stress tensor components:

$$\overline{\tau}_{ij} = \mu\left(\frac{\partial\overline{u}_i}{\partial x_j} + \frac{\partial\overline{u}_j}{\partial x_i}\right)$$

Finally, if the *Reynolds averaging* is applied to the mass and energy balance equations, we obtain, as in laminar flow, a system of equations which is not closed. Closure requires use of some approximations, which take form prescribing the Reynolds-stress tensor and the turbulent scalar flux.

- *Large Eddy Simulations, LES*

  As we have said before, turbulent flows contain a wide range of eddy sizes. The large eddies are much energetic than the small ones, so the transport of the conserved propierties is much effective. In *Large Eddy Simulations*, big structures are solved, while the small ones are modeled. These simulations are three dimensional, time dependent and expensive.

  The velocity field is filtered to have only the large scale components of the total field. The filtered velocity is defined by

$$\overline{u}_i(x) = \int G(x, x^{'}) u_i(x^{'}) dx^{'} \tag{3.12}$$

  where $G(x, x^{'})$ is the filter kernel and is a localized function.

  When the *Navier-Stokes equations* are filtered, one obtains:

$$\frac{\partial(\rho\overline{u}_i)}{\partial t} + \frac{\partial(\rho\overline{u_i u_j})}{\partial x_j} = -\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right)\right] \tag{3.13}$$

  It is important to note that

$$\overline{u_i u_j} \neq \overline{u}_i \overline{u}_j$$

  so, a modeling approximation of the difference between both values is introduced.

$$\tau_{ij}^{s} = -\rho(\overline{u_i u_j} - \overline{u}_i \overline{u}_j) \tag{3.14}$$

  τ_{ij}^{s} is called the *subgrid-scale Reynold stress* in the context of LES. It is related to the momentum flux in the large scales caused by the action of the small scales. The models used to aproximate it are called *subgrid-scale (SGS)* or *subfilter-scale models*.

- *Direct Numerical Simulation, DNS*

  Direct Numerical Simulation is the most accurate method because it resolves Navier-Stokes equations without averaging. The only approximation is numerical discretization, which errors can be estimated. The computed flow field obtained is equivalent to a laboratory experiment.

  In this kind of simulation, to ensure that all significant structures of the turbulence have been captured, the domain of study must be at least as large as the largest turbulent eddy. This scale is called the *integral scale* of the turbulence (L). Also,

the simulation must capture all the kinetic energy dissipation, which takes place on the smallest scales. So, the grid must be no larger than the *Kolmogorov scale* (η). Normally, DNS is used in simple geometries and low Reynolds numbers, since the number of grid points is limited by the processing speed and memory of the computer. Moreover, the results obtained are very detailed, but it is too expensive to use it very often. So, it could be conclude that the major role of *Direct Number Simulation* would be as a research tool to understand qualitatively the physics of the flow.

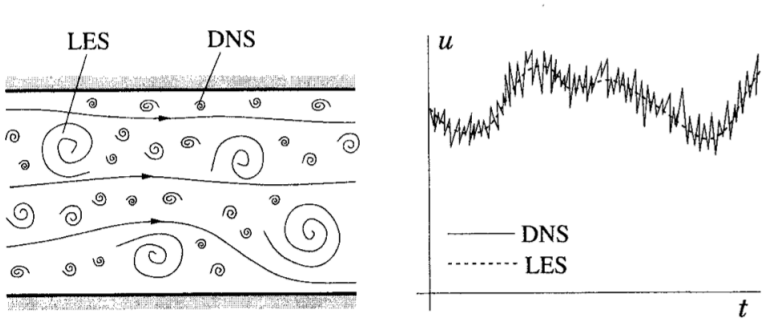


Figure 3.7: Comparison between LES and DNS methods

3.1.5 Boundary conditions

The exact solution in a problem is subjected to the *boundary conditions*. The solution must satisfy the governing equations and these conditions in the boundaries. When we impose *boundary conditions* on a problem with differential equations, it will have a unique solution. So, it is very important to establish them correctly.

Normally, we define a type of boundary condition, with a value and a location. Usually, they are *inlet*, *outlet*, *impermeable walls*, *symmetry planes* and *specified preassure*. It is crucial that the boundary conditions have a physical sense. There are different types of boundary conditions, like *Dirichlet* (**??**) or *Neumann* (**??**) conditions, among others.

$$y = f \tag{3.15}$$

$$\frac{\partial y}{\partial x} = f \tag{3.16}$$

3.2 Solver

Sedondly, *Solver* will be performed. In the *Preprocessing*, we have defined the model of the problem, which will be solved numerically.

There are various numerical methods to obtain an approximate solution of the governing equations.

- ***Finite Differences Methods, FDM*** are numerical methods used to solve differential equations by approximating the derivatives with finite differences. The model equation can be written as

$$\frac{\partial(\rho\phi)}{\partial t} + \frac{\partial(\rho u_j \phi)}{\partial x_j} - \frac{\partial}{\partial x_j}\left(\Gamma \frac{\partial \phi}{\partial x_j}\right) - q_\phi = 0 \tag{3.17}$$

  where ρ, u_j, Γ and q_ϕ are known and ϕ is the conservative variable we want to obtain. After discretizing the domain using a mesh, at each grid point is applied the differential equation, which is approximated in terms of the nodal values of the functions. The result is a system of linear algebraic equations, one per each node, in which the variable values at the grid points are unknowns. This method can be used in any type of grid, but it is more commonly applied to structured grids (Figure **??** (a)). Also, it is important to have a special care with the boundary points. If a Dirichlet boundary condition is provided, there is no need of an equation, but if the boundary condition is a Neumann condition, which involves a derivate, it must be discretized too.

  The main idea behind *Finite Difference Methods* is based on the definition of a derivative.

$$\left(\frac{\partial \phi}{\partial x}\right)_{x_i} = \lim_{\Delta x \to 0} \frac{\phi(x_i + \Delta x) - \phi(x_i)}{\Delta x} \tag{3.18}$$

  The first derivative is defined as the slope of the tangent to a curve $\phi(x)$ at a point x_i. This is the exact solution, but it can be approximated in by the slope of a line passing through two nearby points, as we can see in the figure **??**.

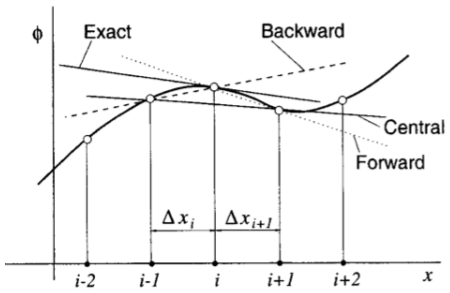


Figure 3.8: Exact derivative and its approximations

These approximations will be more accurate if the grid is refined, i.e. the points are closer to x_i.

To approximate the partial derivates are used some methods as *Taylor series expansion* or *polynomial fitting*.

– **Taylor Series Expansion**

Any continuous fuction $\phi(x)$ can be expressed as a Taylor series

$$\phi(x) = \phi(x_i) + (x - x_i)\left(\frac{\partial\phi}{\partial x}\right)_i + \frac{(x-x_i)^2}{2!}\left(\frac{\partial^2\phi}{\partial x^2}\right)_i + \\ + \frac{(x-x_i)^3}{3!}\left(\frac{\partial^3\phi}{\partial x^3}\right)_i + ... + \frac{(x-x_i)^n}{n!}\left(\frac{\partial^n\phi}{\partial x^n}\right)_i + H \tag{3.19}$$

where H is the higher order terms. If we replace x by x_{i+1} or x_{i-1} in the equation **??**, we will obtain the variable values at these points in terms of the variable and its derivates. One can approximate the first and higher derivatives at the point x_i in terms of neighboring points. We obtain the first derivative at x_i using the **??** for x_{i+1}. It is known as *Forward Difference Scheme, FDS.*

$$\left(\frac{\partial\phi}{\partial x}\right)_i = \frac{\phi_{i+1} - \phi_i}{x_{i+1} - x_i} - \frac{x_{i+1} - x_i}{2}\left(\frac{\partial^2\phi}{\partial x^2}\right)_i - \frac{(x_{i+1} - x_i)^2}{6}\left(\frac{\partial^2\phi}{\partial x^2}\right)_i + H \tag{3.20}$$

Backward Difference Scheme, BDS uses Taylor series for $\phi(x_{i-1})$ to obtain the first and higher derivates. The result is

$$\left(\frac{\partial\phi}{\partial x}\right)_i = \frac{\phi_i - \phi_{i-1}}{x_i - x_{i-1}} - \frac{x_i - x_{i-1}}{2}\left(\frac{\partial^2\phi}{\partial x^2}\right)_i - \frac{(x_i - x_{i-1})^2}{6}\left(\frac{\partial^3\phi}{\partial x^3}\right)_i + H \tag{3.21}$$

Finally, using the equation **??** at x_{i-1} and x_{i+1} to derive the expression of the first and higher derivatives, we carried out *Central Differences Scheme, CDS.*

$$\left(\frac{\partial\phi}{\partial x}\right)_i = \frac{\phi_{i+1} - \phi_{i-1}}{x_{i+1} - x_{i-1}} - \frac{(x_{i+1} - x_i)^2 - (x_i - x_{i-1})^2}{2(x_{i+1} - x_{i-1})}\left(\frac{\partial^2\phi}{\partial x^2}\right)_i - \\ - \frac{(x_{i+1} - x_i)^3 - (x_i - x_{i-1})^3}{6(x_{i+1} - x_{i-1})}\left(\frac{\partial^3\phi}{\partial x^3}\right)_i + H \tag{3.22}$$

The approximations of the derivatives can be simplified by delating some terms from the right hand side. Nevertheless, these terms are called *truncation error* because they are directly related on the accuracy of the solution. This error is known and it can be decreased when the grid is refined.

– **Polynomial Fitting**

Another technique to obtain approximations for derivatives is to define a function with an interpolation curve and differentiate it. For example, if we define a first order polynomial, we obtain

$$\phi(x) = a + bx, \quad \frac{\partial\phi}{\partial x} = b \tag{3.23}$$

Setting the origin of the coordinate system at x_i, we define $\Delta x_{i+1} = x_{i+1} - x_i$ and $\Delta x_i = x_i - x_{i-1}$. Then, solving the system of equations, we get

$$\left(\frac{\partial\phi}{\partial x}\right)_i = \frac{\phi(x_{i+1}) - \phi(x_i)}{\Delta x_{i+1}} \tag{3.24}$$

which is the first order approximation evaluated in Δx_{i+1}.

– **Compact Schemes**

These schemes are used for uniform grids. They are similar to the *polynomial fitting*, but instead of using only the variables values to obtain the coefficients of the polynomial, also the derivatives at some points are used.

We are going to obtain the approximation for the first derivate using a fourth degree polynomial.

$$\phi(x) = a_0 + a_1(x - x_i) + a_2(x - x_i)^2 + a_3(x - x_i)^3 + a_4(x - x_i)^4 \tag{3.25}$$

If we differentiate it, we get

$$\frac{\partial \phi}{\partial x} = a_1 + 2a_2(x - x_i) + 3a_3(x - x_i)^2 + 4a_4(x - x_i)^3 \tag{3.26}$$

After replacing x by x_i, x_{i+1} and x_{i-1} in the equations **??**, **??** and doing some rearrangements, we obtain the following equation

$$\left(\frac{\partial \phi}{\partial x}\right)_i = -\frac{1}{4}\left(\frac{\partial \phi}{\partial x}\right)_{i+1} - \frac{1}{4}\left(\frac{\partial \phi}{\partial x}\right)_{i-1} + \frac{3}{4}\frac{\phi_{i+1} - \phi_{i-1}}{\Delta x} \tag{3.27}$$

This equation will be applied to each node, obtaining a tridiagonal system. A general expression for the *compact centered schemes* is

$$\alpha\left(\frac{\partial \phi}{\partial x}\right)_{i+1} + \left(\frac{\partial \phi}{\partial x}\right)_i + \alpha\left(\frac{\partial \phi}{\partial x}\right)_{i-1} = \beta\frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} + \gamma\frac{\phi_{i+2} - \phi_{i-2}}{4\Delta x} \tag{3.28}$$

where α, β and γ are defined. Secondly the choice of these parameters, we will have the second and fourth order CDS and fourth and sixth order Padé schemes.

- ***Finite Volume Method, FVM*** is a numerical method that uses the integral form of the conservation equations. The model equation, for a steady state, is

$$\int_S \rho\phi\mathbf{v}\cdot\mathbf{n}dS - \int_S \Gamma\nabla\phi\cdot\mathbf{n}dS = \int_V q_\phi dV \tag{3.29}$$

In this case, the solution domain is subdivided into a finite number of control volumes (CV) as we can see in the figure **??** (b) as cells. The conservation equations are applied to each CV. The node is located at the center of the CV, in which the variable values are going to be calculated. Then, one obtains an algebraic equation for each CV. The boundary conditions are assigned to the boundary surface of the computational domain. This method can be applied to any type of grid, even to complex geometries. In addition, it is a conservative method by construction because the surface integrals over inner CV faces, which represent the convective and diffusive fluxes, cancel out, obtaining the global conservation equation of the computational domain as a whole.

On the one hand, to obtain an numerical solution, the surface integral is approximated as a sum of fluxes over each CV faces.

$$\int_S f dS = \sum_k \int_{S_k} f dS \tag{3.30}$$

where f is the normal component vector to the surface, which represents the convective/diffusive flux. The simplest approximation of the surface integral is the midpoint rule, but there are others like trapeziod rule. In the midpoint rule, the flux is approximated by the value at the cell center surface, as we can see in the figure **??**.

$$\int_S f dS = \overline{f}_e S_e \approx f_e S_e \tag{3.31}$$

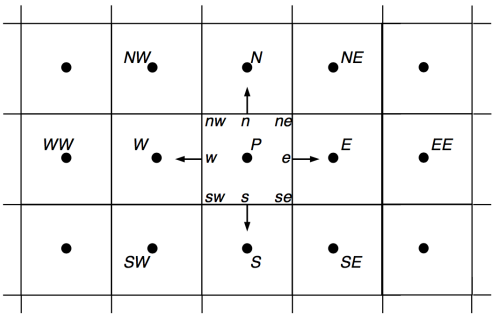


Figure 3.9: Convective/Diffusive fluxes over a control volume

On the other hand, the volume integral is approximated as

$$\int_V q dV = \overline{q} \nabla V \approx q_P \nabla V \tag{3.32}$$

where q_P is the value at the cell center, which is easily calculated. It can be a constant or vary linearly in the CV. An approximation of higher order requires the values of q at more locations. These values can be obtained by interpolation.

As the flux in the cell surfaces are unknown, we must interpolate them. Both fluxes, convective, $f^c = \rho\phi\mathbf{v} \cdot \mathbf{n}$, and diffusive, $f^d = \Gamma grad\phi \cdot \mathbf{n}$, involve the product of several variables. To calculate the value of ϕ and its grandient normal to the cell faces, which are unknown, they must be expressed in terms of nodal values. To carry throught it, the most commonly interpolation schemes are

– *Upwind Differencing Scheme (UDS)*

  This method takes the value of ϕ in the cell center upstream or downstream of the surface, depending on the flow direction.

$$\phi_e = \begin{cases} \phi_P & \text{if } \mathbf{v} \cdot \mathbf{n}_e > 0 \\ \phi_E & \text{if } \mathbf{v} \cdot \mathbf{n}_e < 0 \end{cases} \tag{3.33}$$

This is equivalent to using a backward or forward difference for the first derivative. This method never yield oscillatory solutions because it is numerically diffusive.

– *Linear Interpolation (CDS)*

Another solution to obtain the value of ϕ at a face center is to perfom a linear interpolation between the two nearest nodes.

$$\phi_e = \phi_E \lambda_e + \phi_P (1 - \lambda_e) \tag{3.34}$$

where λ_e is the linear interpolation factor, defined as

$$\lambda_e = \frac{x_e - x_P}{x_E - x_P} \tag{3.35}$$

This scheme is second order accurate for uniform and non-uniform meshes and may produce oscillations, as all schemes of higher order than one. The approximation of the gradient in e can be written as

$$\left(\frac{\partial \phi}{\partial x}\right)_e \approx \frac{\phi_E - \phi_P}{x_E - x_P} \tag{3.36}$$

The CDS is not based in privilaged directions like the upwind scheme.

– *Quadratic Upwind Interpolation (QUICK)*

In this case, the variable ϕ is approximated by a parabola rather than a straight line. Then, we will need to use data from one point more. To choose the third point, we will take W if the flow is from P to E ($u_x > 0$) or EE if $u_x < 0$. The formula is

$$\phi_e = \phi_U + g_1(\phi_D - \phi_U) + g_2(\phi_U - \phi_{UU}) \tag{3.37}$$

where D, U and UU are the downstream, the first upstream and the second upstream node, respectively. The coefficients g_1 and g_2 can be expressed as

$$g_1 = \frac{(x_e - x_U)(x_e - x_{UU})}{(x_D - x_U)(x_D - x_{UU})} \tag{3.38}$$

$$g_2 = \frac{(x_e - x_U)(x_D - x_e)}{(x_U - x_{UU})(x_D - x_{UU})} \tag{3.39}$$

This method is more complex than CDS because it takes into account one node more.

- ***Finite Elements Method, FEM*** is similar to the FV method. In this case, the domain is discretized in control volumes too, that normally are unstructured. The main difference is that the equations are multiplied by a *weight function* before being integred over the whole domain. The simplest FEM solution is approximated by a linear function, to guarantee the continuity of the solution across the element boundaries. Usually, the weight fuctions has the same form. These functions are substituted into the weigthed integral of the conservation form and it is obtained a set of non-linear algebraic equations.

Furthermore, if we consider an unsteady flow, we will have to discretize the temporal term. The conservation equation can be written in a derivative forme as

$$\frac{\partial(\rho\phi)}{\partial t} = -div(\rho\phi\mathbf{v}) + div(\Gamma grad\phi) + q_\phi = f(t,\phi(t)) \tag{3.40}$$

where the function $f(t,\phi)$ is the sum of convective, diffusive and source terms, all of which are discretized as we have explained before at one or more time instants. There are two main methods to discretize the partial derivative in time.

- Explicit Method

  The simplest method is the *explicit Euler method*, which we are going to explain. In this method, the solution at the node is directly calculated at the new time level t_{n+1} because the terms of the equation are evaluated at previous times t_n, in which the variable values are already known.

  In order to unsderstand this method, we are going to consider a 1D flux, with constant velocity, properties and no source terms. The equation **??** becomes to

  $$\frac{\partial(\rho\phi)}{\partial t} = -u\frac{\partial\phi}{\partial x} + \frac{\Gamma}{\rho}\frac{\partial^2\phi}{\partial x^2} \tag{3.41}$$

  Then, we are going to approximate the derivatives using CDS. The variable value at the new time instant is

  $$\phi_i^{n+1} = \phi_i^n + \left[-u\frac{\phi_{i+1}^n - \phi_{i-1}^n}{2\Delta x} + \frac{\Gamma}{\rho}\frac{\phi_{i+1}^n + \phi_{i-1}^n - 2\phi_i^n}{(\Delta x)^2}\right]\Delta t \tag{3.42}$$

  The *Courant number* is a dimensionless parameter, which is defined as the ratio between the time step, Δt, and the caracteristic convection time, $u/\Delta x$. The *explicit method* is *unconditionally unstable*, so we need to impose some restrictions on the time step to guarantee the stability. When the diffusion is negligible, we will use the *Courant-Friedrichs-Lewy condition (CFL)*,

  $$\Delta t < \frac{\Delta x}{|u|} \tag{3.43}$$

  where u is the flow speed, Δt the time step and Δx the spatial length. However, when convection is insignificant, we use the following condition

  $$\Delta t < \frac{\rho(\Delta x)^2}{2\Gamma} \tag{3.44}$$

- Implicit Method

  In this case, we are going to study the *Implicit Euler Method*. In an *implicit method*, the terms of the equation are evaluated at the new time level t_{n+1}, in which the values are unknown. Hence, to get the solution it will be necessary to solve an algebraic system of equations iteratively.

  $$\phi_i^{n+1} = \phi_i^n + \left[-u\frac{\phi_{i+1}^{n+1} - \phi_{i-1}^{n+1}}{2\Delta x} + \frac{\Gamma}{\rho}\frac{\phi_{i+1}^{n+1} + \phi_{i-1}^{n+1} - 2\phi_i^{n+1}}{(\Delta x)^2}\right]\Delta t \tag{3.45}$$

Some advantage of the *Implicit Method* are the unconditional stability of the temporal scheme and the possibility to choose the step time in function of the time scale. The disadvange is the high computational cost due to the iterative method used to solve the system of equations at each time step.

Furthermore, once the equations are linearized, they have to be solved. In FLUENT, we can choose between two types of *solvers*:

- **Density-based solver**
  On the one hand, density, ρ, is obtained from the *continuity equation*. On the other hand, pressure, p, is calculated from the *state equation*. *Navier-Stokes and species equations* are solved simultaneously. Then, turbulence and other scalar equations are resolved sequentially. *Density-based solver* uses an iterative process to solve the governing equations until the convergence of the solution is reached and it is recommended for flows with high Mach number.

- **Pressure-based solver**
  This solver uses an algorithm, in which the pressure, p, is obtained by the combination of the *continuity* and *momentum equations*. Then, the density, ρ, is calculated from the *state equation*. The velocity, v, is corrected by the pressure to satisfy the constrait of *mass conservation equation*. Pressure-Velocity coupling is very important only in the *Pressure-based solver*. So, FLUENT provides different methods to solve it: SIMPLE, SIMPLEC, PISO and Fractional Step (FSM). All these methods are based on the predictor-corrector approach and use the pressure-based segregated algorithm. On the one hand, SIMPLE and SIMPLEC are recommended for steady-state calculations. On the other hand, PISO is used in transient calculations, while Fractional Step is better to time-dependent flows with non-iterative time advancement option. In summary, as the governing equations are non linear and coupled, they are solved with an iterative method until the solution is converged. This kind of solver is suggested to low and moderate Mach numbers.

In addition, when you are solving a system of coupled equations in a linearized and sequential way, you should control the *Under-relaxation factor, URF*. The algebraic equation on the nth iteration may be written as

$$A_P \phi_P^n + \sum_l A_l \phi_l^n = Q_P \tag{3.46}$$

where Q is the source term and contains all the terms that not depend on ϕ^n. This equation is linear and it is solve iteratively. *Under-relaxation factor*, α_ϕ, allows us to control the change of the variable ϕ^n to avoid instabilities.

$$\phi^n = \phi^{n-1} + \alpha_\phi(\phi^{new} - \phi^{n-1}), \quad 0 < \alpha_\phi < 1 \tag{3.47}$$

where ϕ^{new} is the result of the equation **??**.

The *Under-relaxation factor* is important because it has a direct effect on the convergence process. If α_φ is small, the convergence process will be stronger but slower too.

However, if α_φ has a large value, the convergence process will speed up, but it may be instable and diverge. A good strategy is to use a small under-relaxation factor in the early iterations and increase it as convergence is approach.

Last but not least, we must pay attention to the *Residuals*. *Residuals* are the sum of the new and previous value of the conserved variables at the end of each iteration. This parameter give us an idea of how much the solution is converged and is computed and stored in a file. On a computer with infinite precision, these residuals will go to zero as the solution converges. However, the residuals value normally decreases until a constant one in which it is maintained. In general, the normalized expression of the *Residuals* is:

$$\bar{R^\phi} = \frac{R^\phi_{iteration\ N}}{R^\phi_{iteration\ M}}$$

where M is the number of iterations after which is obtained the maximum residual value. M can be set by you in the Residuals Monitors and by default is $M = 5$. This expression show us how much the residual has decreased during the calculations.

3.3 Postprocessing

Finally, *Postprocessing* is the last step in CFD simulation. After obtaining the results of the variables, it is important analyse them and try to explain their meaning. We can make XY plots, graphics or animations to help us to understand the solution and verify their physical sense.

Chapter 4

Benchmark description

In this chapter, we are going to explain the process carried out to define and solve computationally the problem investigated in this Thesis.

4.1 Physical and computational set-up

As we have said before, the aim of this project is to research the effect of an inadequate mesh in the propagation of the error to the final solution. The problem of flow past a square cylinder has been chosen due to the fact that its importance in some fields, there are many experimental investigations to compare the results and its computational set-up is not too complicated. In this section, we describe the main parameters defined and, in the next one, we will be focused only in the central part of this Thesis, the mesh.

Geometry

To define the geometry showed in the figure **??**, we have used the *DesignModeler* tool, included in ANSYS.

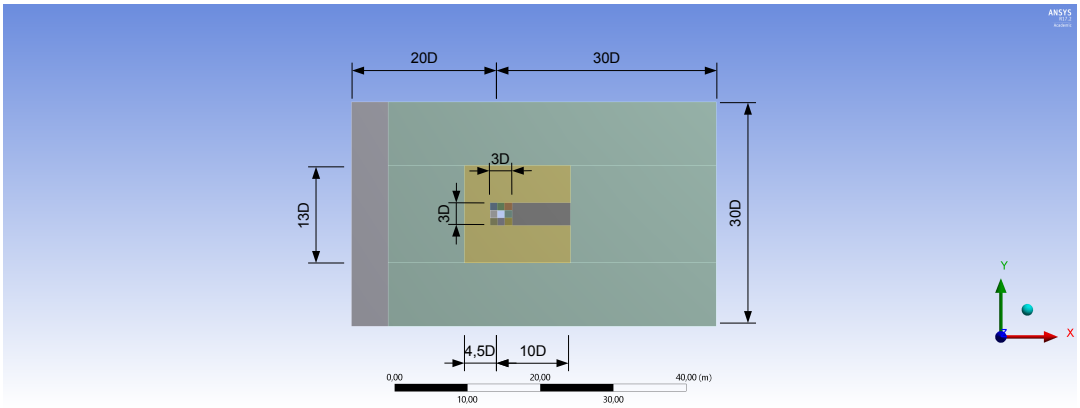


Figure 4.1: Geometry

In this picture, the dimensions of the different domains are included. This geometry is 2D and symmetrical in the x-axis. It is simple but suitable to represent the case of study.

Additionally, in the figure **??** we differenciate the domains from out to inside as : 1) *outer ring*, 2) *middle C*, 3) *wake* and 4) *inner ring*, around which we are going to study how the flow behaves.

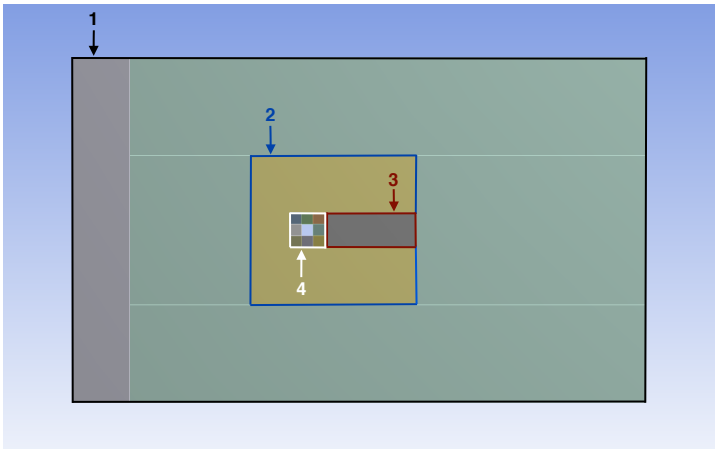


Figure 4.2: Domains

After the geometry, it is defined the mesh, but it will be explained in detail in the Section 4.2.

Model and flow attributes

To carry through this project, we use a type of *RANS Model*. As we have said in the Chapter 3, the set of equations is not closed. Thus, to do it we must define a *turbulence model*. In particular, we have selected the $k-\epsilon$ *model*, where k is the kinetic energy and ϵ the dissipation, both parameters of the turbulence.

In essence, turbulence can be characterized by its kinetic energy (k) or velocity ($q = \sqrt{2k}$) and a length scale (L). The turbulent kinetic energy is defined by the following expression:

$$k = \frac{1}{2}\overline{u_i' u_i'} = \frac{1}{2}(\overline{u_x' u_x'} + \overline{u_y' u_y'} + \overline{u_z' u_z'}) \tag{4.1}$$

Due to the complexity related to the description of the turbulence quantities suggests that one might use partial differential equations to compute them. The equation for the *turbulent kinetic energy, k,* is written as

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho \bar{u}_j k)}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\mu \frac{\partial k}{\partial x_j}\right) - \frac{\partial}{\partial x_j}\left(\frac{\rho}{2}\overline{u_j' u_i' u_i'} + \overline{p' u_j'}\right) - \rho \overline{u_i' u_j'} \frac{\partial \bar{u}_i}{\partial x_j} - \mu \overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial u_i'}{\partial x_k}} \tag{4.2}$$

The last term represents the product of the density (ρ) and the dissipation (ϵ), when the turbulence energy is converted into internal energy. The second term on the right-hand

side of the expression **??** is known as *turbulent diffusion*, which can be modeled like

$$-\left(\frac{\rho}{2}\overline{u_j' u_i' u_i'} + \overline{p' u_j'}\right) \approx \frac{\mu_t}{\sigma_k}\frac{\partial k}{\partial x_j} \tag{4.3}$$

where σ_k is the turbulent Prandtl number and its value approaches unity.
The third term of the right-hand side represents the rate of production of turbulent kinetic energy by the mean flow. If we estimate the Reynolds stress using the eddy viscosity, the equation results

$$P_k = -\rho\overline{u_i' u_j'}\frac{\partial \overline{u}_i}{\partial x_j} \approx \mu_t\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right)\frac{\partial \overline{u}_i}{\partial x_j}. \tag{4.4}$$

As mentioned above, two equations are required to close the set of equations. In turbulence, it is widely known that energy is tranferred from the largest scales to the smallest until it is dissipated. This relationship can be formulated.

$$\epsilon \approx \frac{k^{3/2}}{L} \tag{4.5}$$

So, the equation of the dissipation (**??**) allows us to complete the model.

$$\frac{\partial(\rho\epsilon)}{\partial t} + \frac{\partial(\rho u_j \epsilon)}{\partial x_j} = C_{\epsilon 1} P_k \frac{\epsilon}{k} - \rho C_{\epsilon 2}\frac{\epsilon^2}{k} + \frac{\partial}{\partial x_j}\left(\frac{\mu_t}{\sigma_\epsilon}\frac{\partial \epsilon}{\partial x_j}\right) \tag{4.6}$$

In this model, the eddy viscosity is defined like

$$\mu_t = \rho C_\mu \sqrt{k} L = \rho C_\mu \frac{k^2}{\epsilon} \tag{4.7}$$

The commonly values of the parameters are: $C_\mu = 0.09$, $C_{\epsilon 1} = 1.44$, $C_{\epsilon 2} = 1.92$, $\sigma_k = 1.0$ and $\sigma_{epsilon} = 1.3$.

To be exact, we use the *RNG* $k - \epsilon$ model, which is derived from the instantaneous Navier-Stokes equations, but applying a rigorous technique, known as *renormalization group theory*. It results similar to the $k - \epsilon$ equations explained before, but with different constants and additional terms. Also, non equilibrium wall functions are imposed as a condition near the wall.

Furthermore, in this Thesis, we are going to work with a transient air flow, which is transfered with a high Reynolds number ($Re = 2.2 \cdot 10^4$). This flow moves in a 2D space and its dimensionless propierties are defined as:

$$\rho = 1$$

$$\nu = \frac{1}{Re} \approx 4.54 \cdot 10^{-5}$$

Boundary conditions

In this study, we have imposed the next boundary conditions in FLUENT:

Boundary condition	Type	Characteristics
Cylinder	Wall	It is assumed stationary, non-rough wall and viscous flow.
Inlet	Velocity inlet	The only component of the velocity at the inlet is horizontal. Turbulent intensity and length scale are defined as 2% and 0.5, respectively.
Lateral	Symmetry and Periodic	Within periodic type, we impose a traslational motion.
Outlet	Pressure outlet	It is established the Dirichlet condition: $p_{rel} = 0$ and the direction of the flow normal to the boundary. Turbulence intensity and length scale has been imposed equal to those defined in the *inlet boundary condition*.

Table 4.1: Boundary conditions

Solution Methods

In addition, the numerical methods and solver imposed to discretize and resolve the problem suggested in this Thesis are the following:

- **Solver:** Pressure-Based.
- **Pressure-Velocity coupling:** PISO with *Skewness Correction* and *Neighbor Correction* both equal to 2.
- **Spatial discretization**
  - **Gradient:** Least Squares Cell Based.
  - **Pressure:** Second Order.
  - **Momentum:** QUICK.
  - **Turbulent Kinetic Energy:** QUICK.
  - **Turbulent Dissipation Rate:** QUICK.
- **Transient Formulation:** Second Order Implicit (Euler method).

Convergence and Data Save

Finally, the convergence ratio is defined by the *Residuals*. We edit the monitor of residuals, where it is possible to select which variables you would like to control the convergence and to define an absolute criteria for the residuals. Normally, this value is small enough (0.001 in all) to avoid errors induced by the solver.

As a part of the study, we must compare the results obtained between them. So, we have to save the aerodynamics coefficients (*lift*, *drag* and *moment*) in different files for each case. It is convenient to save regularly the results of the simulations. So, we define the number of time steps after which we would like to save the information and the value of one time step.

4.2 Parametrical study set-up

In this section, we are going to explain only how the mesh, which is the central part of this Thesis, has been generated. Concretly, the focus of study will be the *Inner ring-grid*, which has a subsection apart.

Sub-structuring strategy

The mesh has been generated after the creation of the geometry and before the definition of the turbulence model. We use the tool *Meshing* included in ANSYS. The general domain is divided in four section, as we have seen in the figure **??**. Hence, for each part, a concrete type of mesh has been defined:

1. *Outer ring-grid:* All mesh is structured but in the rear part, near the wake, it is unstructured.

2. *Middle C-grid:* We have defined a triangular unstructured mesh.

3. *Wake grid:* The mesh is structured orthogonal.

4. *Inner ring-grid:* As we have said at the start of this section, we are going to study deeply the error propagated to the solution by a low quality mesh. So, to investigate this, we have generated different types of grids for this zone of the domain (Figure **??**).

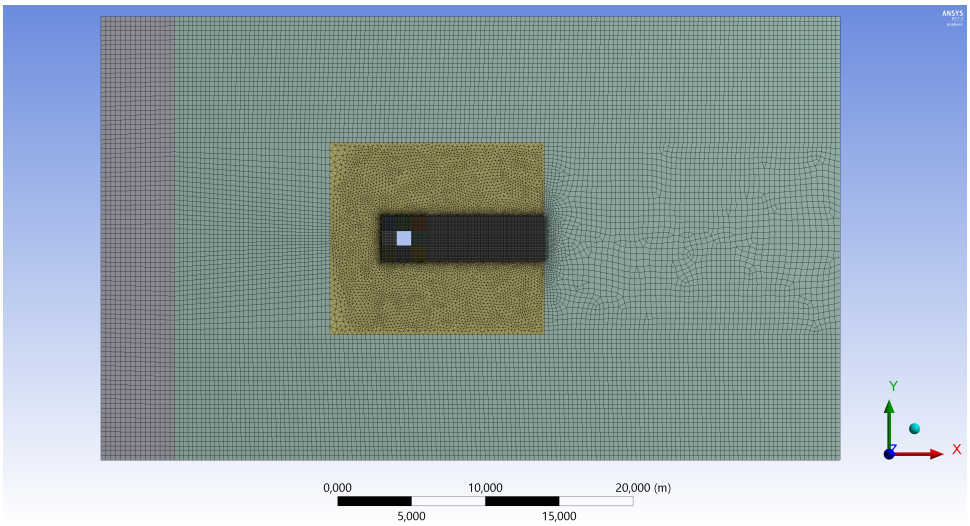


Figure 4.3: Mesh

To create these meshes, we have to impose some conditions that are compiled in the table **??**. We reference each line of the domain with a letter, as it is shown in the figure **??**. Then, we define the number of elements in which we want to divide the edge and the bias type. If it is active this option, we should choose a bias factor that is defined as the total expansion ratio of the cells in this side. In all the cases, we impose a hard behavior option, which means that the size or number of divisions is fixed on the edge and cannot be changed.

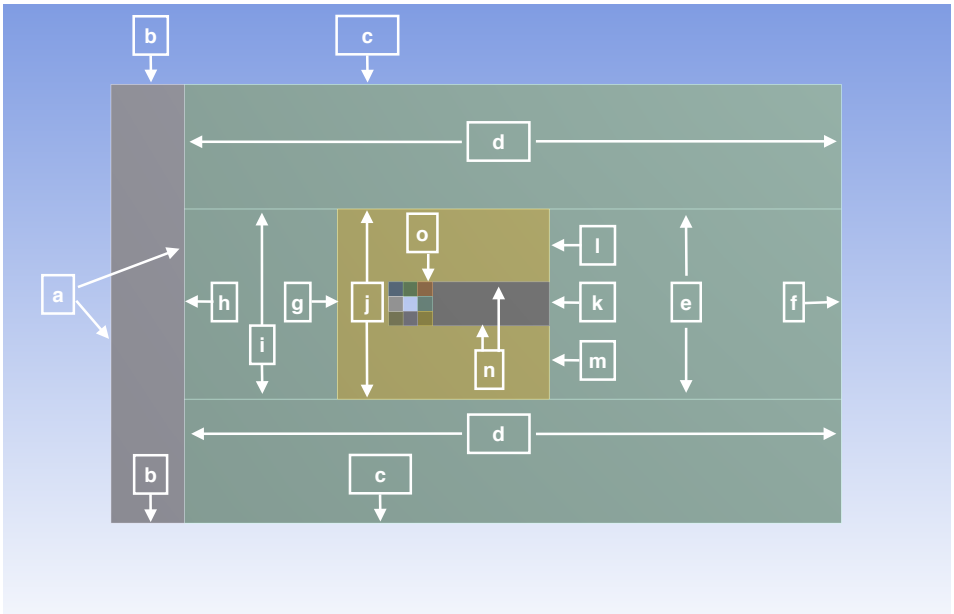


Figure 4.4: References to create the meshes

Label	Bias type	Number of elements
a	no	94
b	no	15
c	no	150
d	no	27
e	no	66
f	no	30
g	... . $fct = 3$	40
h	no	40
i	no	36
j	no	48
k	no	72
l	. .. ... $fct = 3.5$	30
m	... .. . $fct = 3.5$	30
n	no	160
o	no	24

Table 4.2: Characteristics of each edge

On the one hand, the triangular unstructured mesh of *middle C-grid* is characterized by an element size of $0.2\,m$ and the growth rate equal to 1.10.

On the other hand, as we can see in the figure **??**, the section of the *Outer ring-grid* located behind the *Wake grid* is an unstructured mesh. Probably, it is the result of the high variation of the cells size between both parts, which is difficult to resolve for FLUENT with the student version. To minimize this contrast, we have impose an element size of $0.4\,m$, a soft behaviour and a growth rate equal to 1.050.

Inner O-grid

We can see in the Figure **??**, the different types of mesh generated in the *Inner ring*. From right to left side we have the *structured orthogonal*, *structured non orthogonal*, *structured/unstructured* and *unstructured* mesh. The grids of the others sections remain constant. Thereby, the results for the different cases can be compared between them.

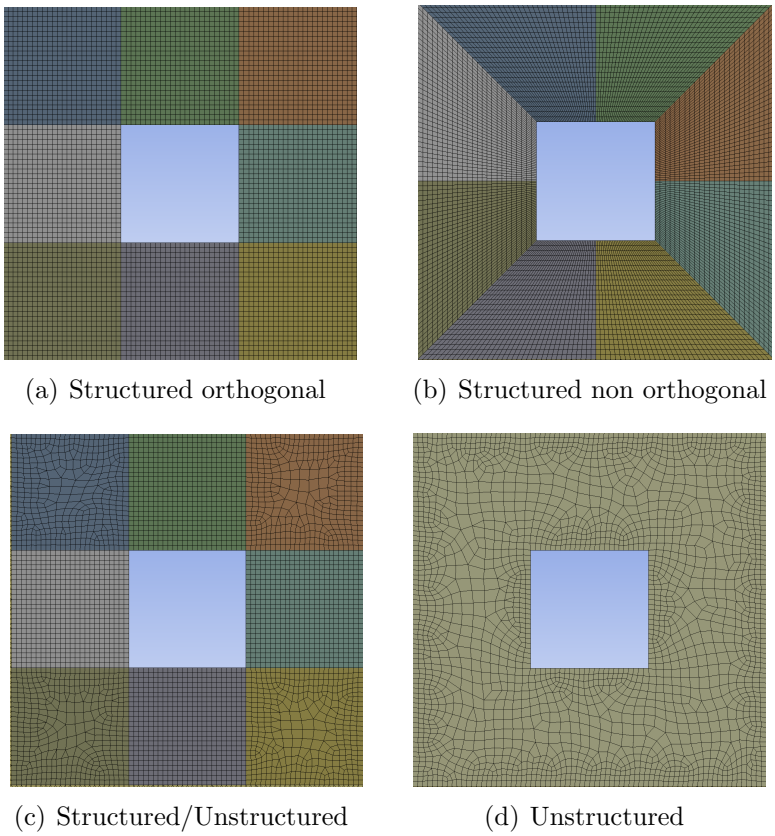
Firstly, for the *structured orthogonal* mesh, the *Inner ring* is divided into eight surfaces. The number of divisions imposed to each edge of these surfaces are 24.

Secondly, the *structured non orthogonal* mesh has been created. In this case, the *Inner ring* is also divided into eight surfaces, but their geometry has changed. To obtain a structured mesh in the *wake*, we have to keep the same number of elements on the left and right sides, 72. So, each edges of the *Inner ring* and the cylinder should have 72 divisions, 36 in each edge of the surfaces. In addition, to mantain the y^+ of the cells near the boundary layer, we impose a number of divisions equal to 24. In this way, we obtain

the desired mesh and it is comparable with others.

Thirdly, the *structured/unstructured* mesh is a combination of both types of grids. We follow the same procedure as in the case of *structured* mesh, but we generate four *structured orthogonal* and four *unstructured* meshes. To carry out the latter, we divide each side into 24 elements, but we impose a no mapped face.

Finally, the *unstructured* mesh is generated. In this case, the external edges of the *Inner ring* are divided into 72 elements and, in the cylinder sides, we impose a number of elements equal to 24. In this way, y^+ is kept constant in all cases because for the first line of cells, the program takes the size of the division that we have defined, and generates a square cell. All the surface is defined as a non mapped face.



(a) Structured orthogonal (b) Structured non orthogonal

(c) Structured/Unstructured (d) Unstructured

Figure 4.5: Inner ring grids

Chapter 5

Results and discussion

After defining and solving the problem using different grids in the *Inner ring*, we can proceed to carry out the *Postprocessing*. In this chapter, the results will be analysed and compared, in order to understand if there is a relationship between mesh quality and the obtaining of inaccurate solutions.

5.1 Mesh quality

The error propagated to the solution can be fundamentally the result of the *truncation error* which appears in the discretization of the equations and the error on account of the *mesh quality*. In essence, the *truncation error* can be determinated knowing the order of numerical scheme used for linearizate the equations. In addition, it is well known that the *mesh quality* is defined by the *aspect ratio*, the *orthogonality* and the *skewness*, as it was expained in the section 3.1.3.

In this Thesis, we investigate deeply the effect of *mesh quality*. Using approximately the same grids for all domains, except for the *Inner ring*, where we define four types of meshes: *structured orthognal*, *structured non orthogonal*, *structured/ unstructured* and *unstructured*.

Mesh	Max. Aspect Ratio	Min. Orthogonality	Max. Skewness
Structured orthogonal	4.85855	4.81664e-01	3.23771e-01
Structured non orthogonal	6.81491	4.81664e-01	4.04156e-01
Structured/Unstructured	5.19822	4.81664e-01	3.50503e-01
Unstructured	6.73504	4.31306e-01	5.68694e-01

Table 5.1: Mesh quality

The table **??** collects the minimum and maximum values of the parameters related to *mesh quality*. The range of the *orthogonality* and *skewness* values goes from 0 to 1. In the first case, when the value is near to zero, means the mesh has a low quality. In

the *skewness* case, the values near to 0 correspond to a high quality. Generally, for the *aspect ratio*, it is best to avoid sudden changes in areas where the flow field exhibit strong gradients. Probably, in all cases, the maximum value obtained for the *aspect ratio* is due to the abrupt change in the cell size between the wake and the rear part of the outer ring. We have defined the wake domain sufficiently long to catch the main information. Hence, we are going to consider the effect of *aspect ratio* negligible in the solution. Later, if we pay attention to the *orthogonality* values, we can see they are similar. This leads us to think that the *non-orthogonality* has an effect in the error of the solution, but it probably will be the same in all meshes and we can not consider it. The only parameter which is different for each grid is the *skewness*.

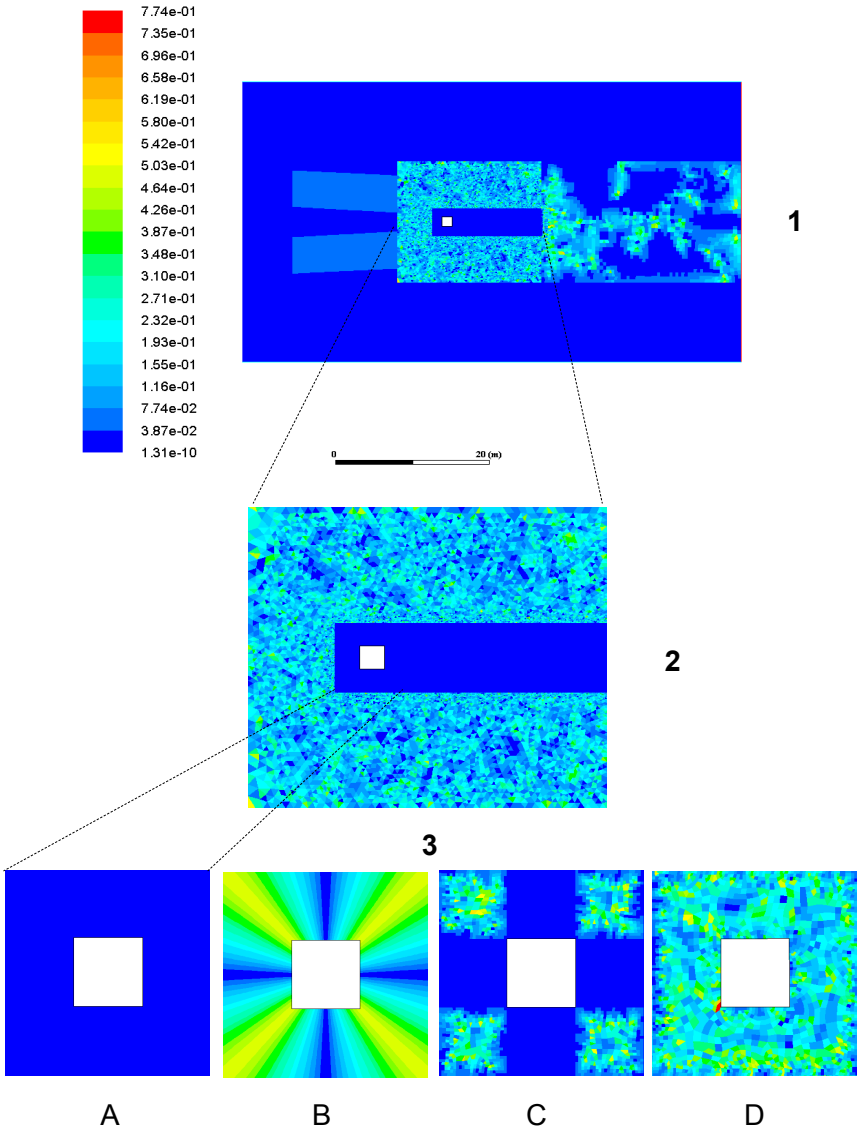
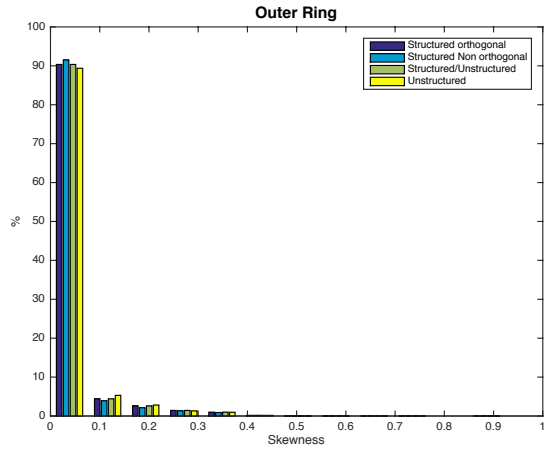


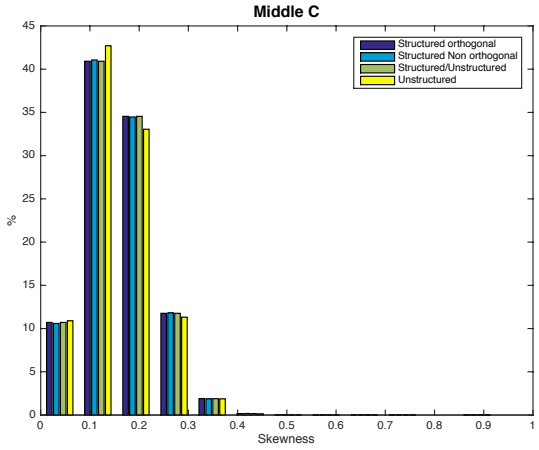
Figure 5.1: Skewness contours. (A) Structured orthogonal; (B) Structured non orthogonal; (C) Structured/Unstructured; (D) Unstructured

In the figure **??**, the various skewness contours have been shown. The entire domain is referenced with the number *1*. The *Middle C* and the *Wake* are indicated with the number *2*. Lastly, the number *3* is related with the different grids of the *Inner ring*. In the upper left side of the picture, we can see the scale of colors used with the corresponding values. On the one hand, the minimum skewness value is $1.31 \cdot 10^{-10}$ and is referred to navy blue color. On the other hand, the maximum value is $7.74 \cdot 10^{-1}$ and is represented by red color. The structured meshes apear completely colored blue. In the *structured non orthogonal* mesh, we observe a variation of colors from navy blue to yellow. It is in the corners of the square-cylinder where there is higher skewness. The *unstructured* meshes present almost all colors in a distributed way.

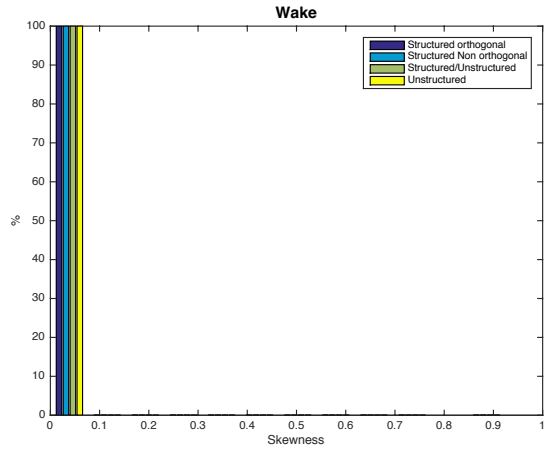
To define which mesh has a larger skewness factor, the percentage data have been collected for each part of the domain in all cases and they have been plotted in the following histograms.



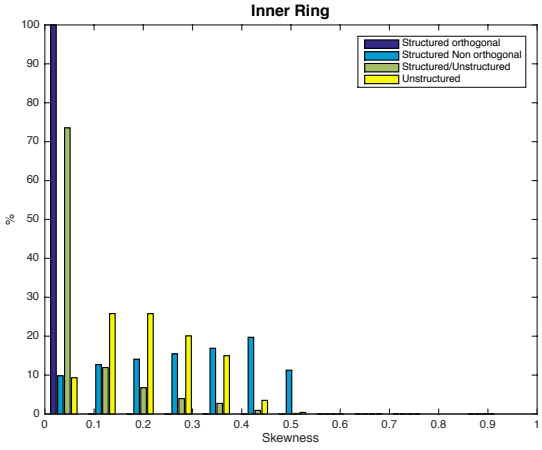
(a) Outer Ring



(b) Middle C



(c) Wake



(d) Inner Ring

Figure 5.2: Histograms

As we can see in the figures **??** (a), (b) and (c), generally, the skewness values concide approximately for the four cases, as we expected. Theoretically, the *outer ring*, *middle C* and *wake* grids should be the same for all cases. However, some variances can appear according to the direction in which FLUENT generates the mesh. Finally, the value of the skewness for the *structured orthogonal*, *structured non orthogonal*, *structured/unstructured* and *unstructured* mesh is different, as it is shown in the figure **??** (d). The worst results are obtained for the *structured non orthogonal* grid. It is because its higher values of skewness (~ 0.4) are related to the largest percentage value of total elements ($\sim 20\%$). In the second place, the *unstructured* grid is located. The highest percentage of elements near to 20% has a skewness value of 0.1. In the case of the *structured/unstructured* mesh, most of the elements have a skewness value approximate to 0.05 because of the effect of the structured meshes. Last but not least, the *structured* mesh have all the elements with the minimum skewness.

5.2 Aerodynamic Coefficients

Lift, *Drag* and *Moment coefficients* are known as the *Aerodynamics Coefficients*. They can be expressed, respectively, as

$$C_L = \frac{L}{\frac{1}{2}\rho V^2 D} \tag{5.1}$$

$$C_D = \frac{D}{\frac{1}{2}\rho V^2 D} \tag{5.2}$$

$$C_M = \frac{M}{\frac{1}{2}\rho V^2 D^2} \tag{5.3}$$

where D is the reference length of the square cylinder.

In a simulation, these coefficients can be calculated by FLUENT. After solving the four cases suggested, one for each *Inner ring-mesh*, we take the data files saved and compare the results obtained. To carry through this task, we plot each *aerodynamic coefficient* for all cases: *structured orthogonal*, *structured non orthogonal*, *structured/unstructured* and *unstructured* mesh, as we can see in the figures **??**, **??** and **??**. We are going to consider as a reference the results of the *structured orthogonal* mesh because it gives the most accurate solutions according with the experimental solutions where $C_L = \pm 1.5$. In all cases, we define the same time step, $\Delta t = 0.02$. To make sure we arrive to a steady flow, we set up a number of iterations equal to 9000, except for the *structured non orthogonal* mesh, in which we impose 12000 iterations because the flow takes longer to be stationary.

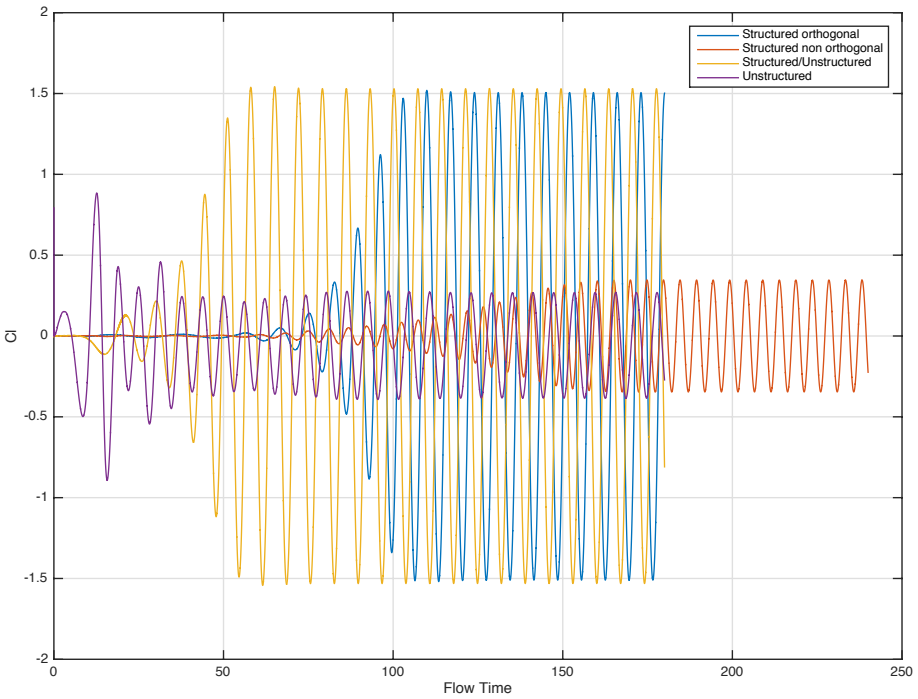


Figure 5.3: Lift coefficient

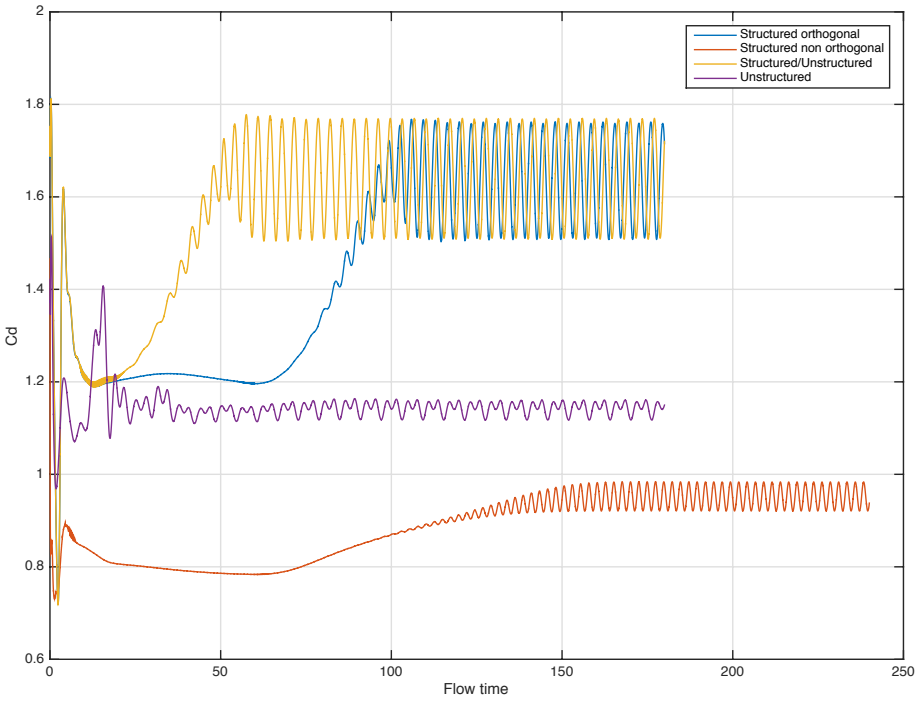


Figure 5.4: Drag coefficient

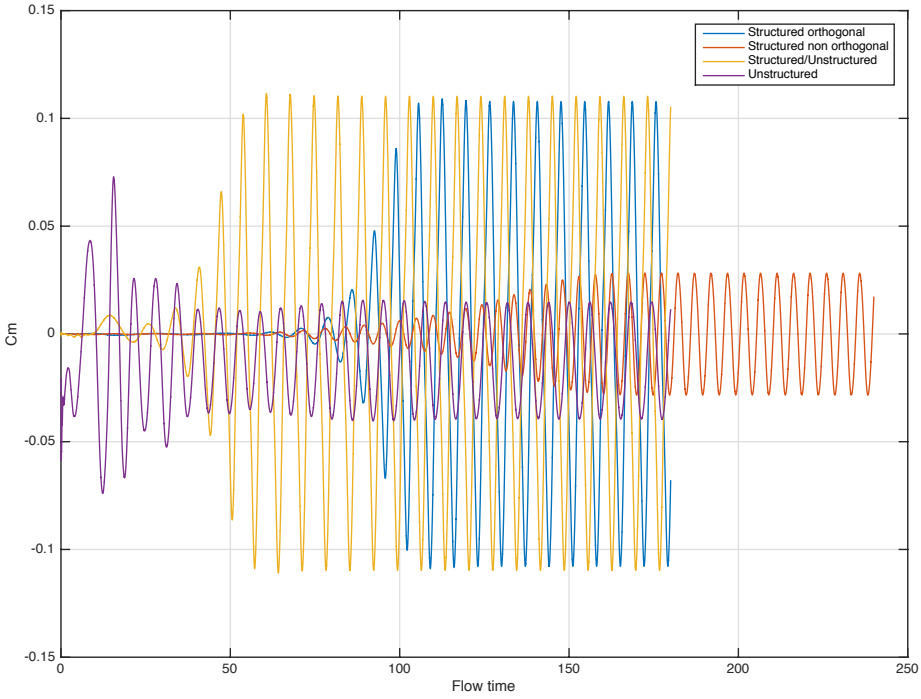


Figure 5.5: Moment coefficient

As we can see in the three previous figures, the most approximate result to the *structured* mesh is obtained by the *structured/unstructured* grid, but it is a bit greater. This accuracy could be the consequence of having a great deal of the mesh structured. Moreover, in this mesh the flow becomes faster steady rather than the *structured* mesh. The *structured non orthogonal* and *unstructured* mesh, both give an inaccurate solution. In the figure **??**, the first one provides a result lower than 1/3 of the correct solution. The result of the *unstructured* grid is lower too and it does not oscillate around zero. Probably, it is due to the asymmetry of the mesh. It occurs the same in the plot of moment coefficient (Figure **??**) but with different values respect to the plot of the lift coefficient. In the figure **??**, we can see the averaged value is different to zero in all cases. Also, the results of the drag coefficient provided by the *structured/unstructured* are the most accurate, while the results of the *structured non orthogonal* and *unstructured* differ a lot of the appropriate solution.

Furthermore, the maximum ($\tilde{C}_L$) and average ($\bar{C}_L$) aerodynamic coefficients have been collected in the following table (**??**). The dimensionless period, T, and the Strouhal number, St, are also in the table. The Strouhal number, for a dimensionless model, is obtained using the expression **??**, but in our case U and B are equal to unity. The equation becomes

$$St = \frac{1}{T} \tag{5.4}$$

It is easy to observe in the table **??** that the average value of the lift and moment coefficients is different from zero when the mesh is *unstructured*. The average drag coefficient

is not zero in all cases.

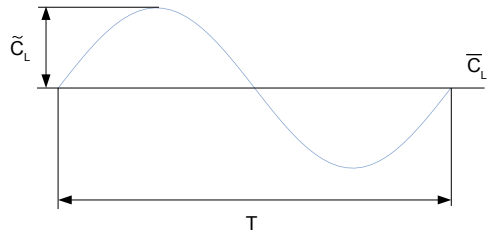


Figure 5.6: Representation of the period, average and maximum coefficient

Mesh	$\overline{C}_L$	$\tilde{C}_L$	$\overline{C}_D$	$\tilde{C}_D$	$\overline{C}_M$	$\tilde{C}_M$	T	St
Structured orthogonal	0	1.506	1.636	1.762	0	0.108	7.04	0.142
Structured non orthogonal	0	0.346	0.952	0.983	0	0.028	4.92	0.203
Structured/Unstructured	0	1.530	1.639	1.770	0	0.110	7.08	0.141
Unstructured	-0.058	0.269	1.139	1.161	-0.0124	0.015	6.12	0.163

Table 5.2: Maximum and averaged coefficients

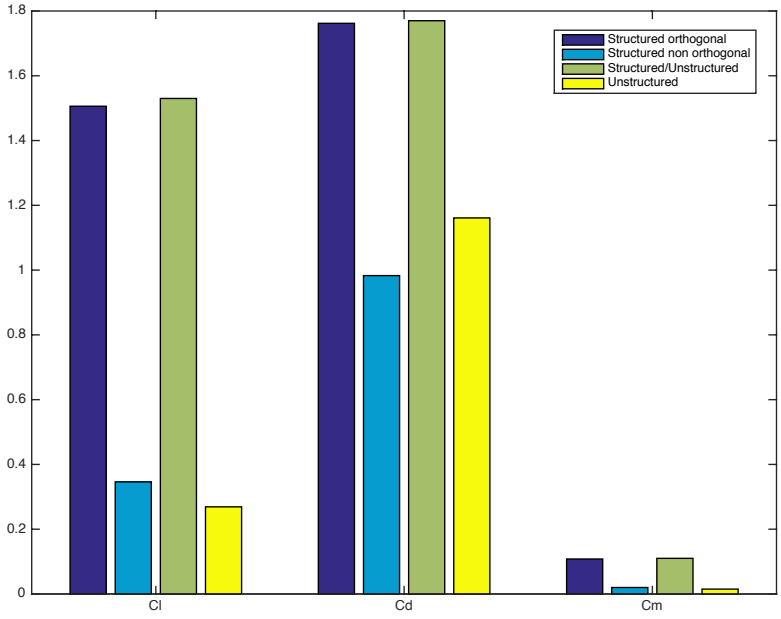


Figure 5.7: Maximum aerodynamic coefficients

In addition, we plot the maximum values of the aerodynamic coefficients in a histogram for the *structured orthogonal*, *structured non orthogonal*, *structured/unstructured* and *unstructured* mesh. In this figure (**??**), it is easy to see the differences between the results. As we said before, *structured orthogonal* and *structured/unstructured* mesh obtain a similar solutions, instead of *structured non otrhogonal* and *unstructured* meshes, which get a lower results.

Finally, we calculate the relative error committed in the maximum values of the lift coefficient. We collect the percentage value of the relative error for each case in the table **??** and observe that the unstructured mesh has the highest value of the relative error, followed by the structured non orthogonal mesh.

$$\epsilon(\%) = \frac{|\tilde{C}_L(structured) - \tilde{C}_L(grid_i)|}{\tilde{C}_L(structured)} \cdot 100 \tag{5.5}$$

Mesh	$\epsilon(\%)$
Structured orthogonal	-
Structured non orthogonal	77.025
Structured/Unstructured	1.594
Unstructured	82.138

Table 5.3: Relative error

5.3 Pressure distributions

Another variable related to the aerodynamic forces is the *pressure coefficient*. In this section, we are going to analyse the variations of *mean pressure coefficient*, $\bar{C}_p$, and *RMSE static pressure coefficient*, $C_{p_{rms}}$, around the upper half of the cylinder surface, as it is showed in the figure **??**.

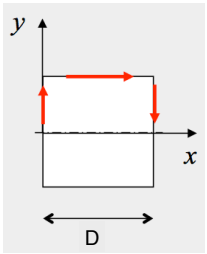


Figure 5.8: Cylinder surface scheme

To get the results of these variables, we simulate only one period, making sure of start in $C_L = 0$ and $\partial C_L/\partial t > 0$. For each case, the period, T, changes. So, keeping the time step, $\Delta t = 0.02$, we modify the number of iterations.

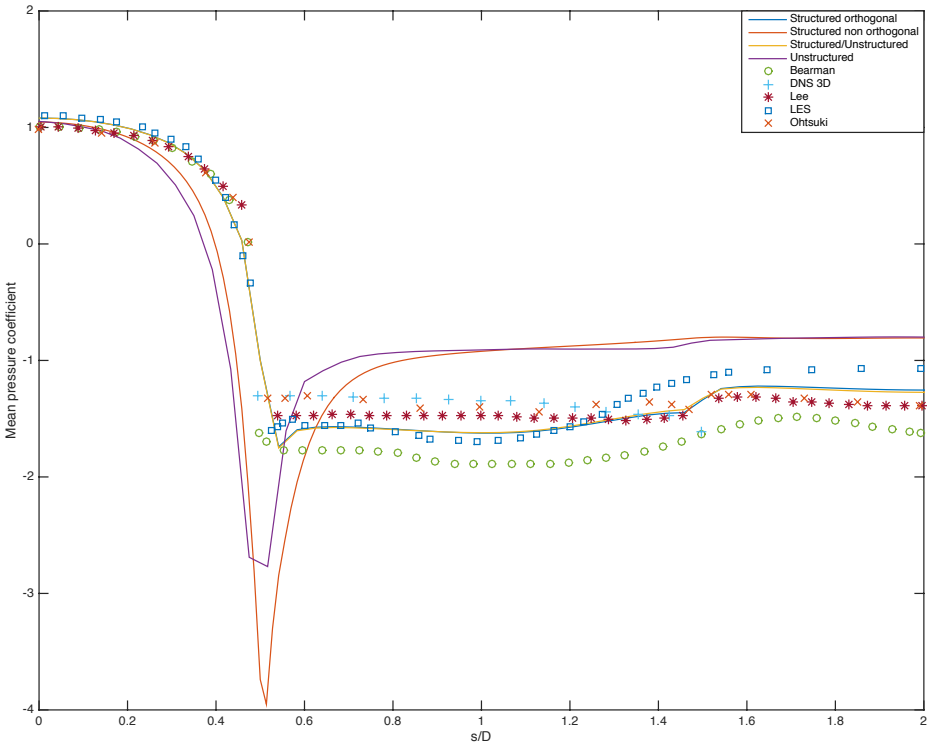


Figure 5.9: Mean pressure coefficient, $\bar{C}_p$

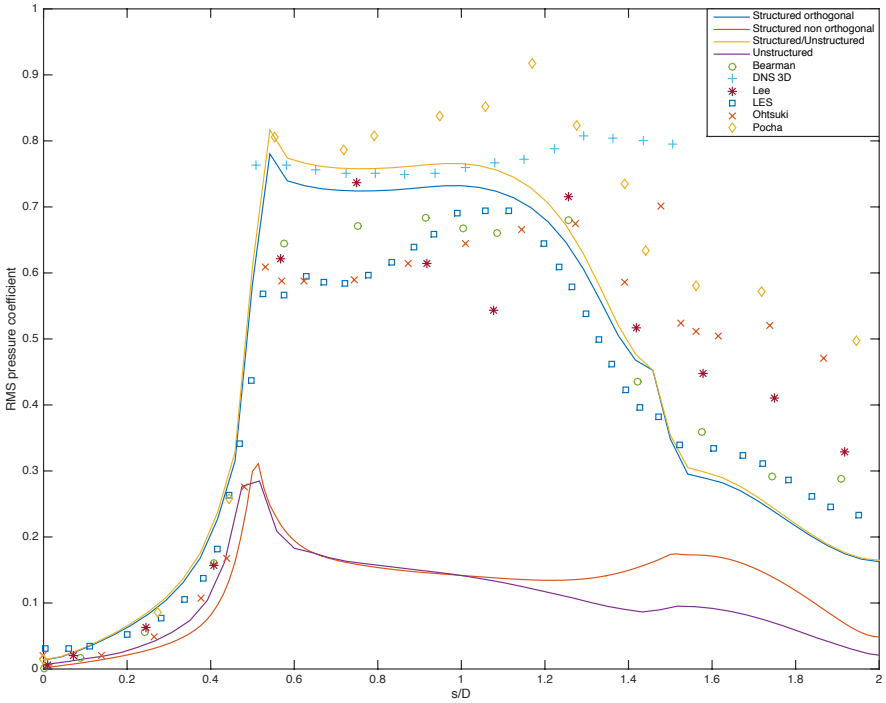


Figure 5.10: RMSE static pressure coefficient, $C_{p_{rms}}$

On the one hand, in the figure **??**, we can see that the *mean preassure coefficient* get using the *structured/unstructured* mesh is almost identical to the reference results obtained from the *structured* mesh. Also, they are close enough to the experimental data collected from the bibliography. In the cases of *structured non orthogonal* and *unstructured* mesh, the results before $s/B = 0.5$ are lower than the correct solution. It means that this grid understimates the *mean preassure coefficient* in these positions. However, after $s/B = 0.5$, the results obtained are overstimated. In all cases, the maximum $\bar{C}_p$ is obtained in the stagnation point, situated in the frontal face, while for the the upper half of the cylinder surface, it is getting a negative $\bar{C}_p$.

On the other hand, we are going to analyse the results of the *RMSE static pressure coefficient*, in which RMSE means root-mean-square error. As we can see in the figure **??**, the results of the *structured/unstructured* mesh are similar to the solutions of the *structured* mesh and theorical data, like in the figure **??**. Also, the *structured non orthogonal* and *unstructured* differ a lot of the reference solution. They understimate the value of the *RMSE static pressure coefficient* across the entire upper half cylinder surface.

5.4 Wake flow

In this last section we are going to study the *mean velocity in the x direction*, $\bar{u}_x$, along a line which divides the square cylinder into two equal parts.

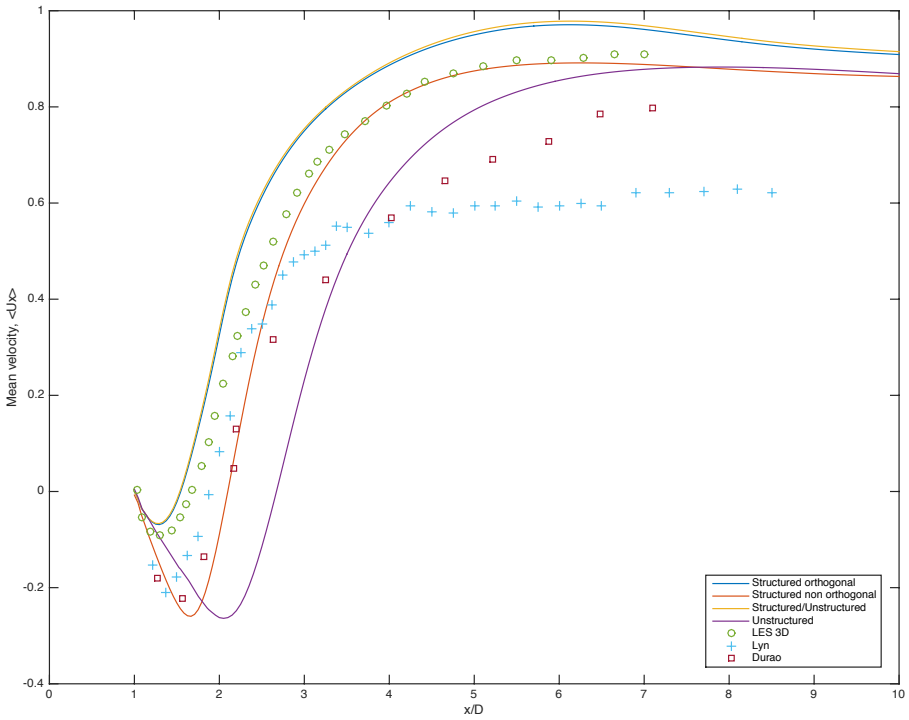


Figure 5.11: Mean velocity in the x direction, $\bar{u}_x$

After the simulation is finished, we plot the data saved starting at the back face of the squared cylinder until $x/D = 10$ and the figure **??** is obtained. Additionally, we plot the experimental data obtained from the bibliography. As in all previous figures, the results obtained by the *structured/unstructured* mesh are the closest to the reference solution (*structured* mesh). In all cases, we can see that the functions firstly decrease to increase later until a maximum. However, secondly the mesh used, this evolution occurs in different points. The curves are moved into the right hand side from the *structured orthogonal* to the *unstructured*, going by *structured non orthogonal* meshes, respectively. As a result of this, the minimum and maximum points are reached in a more distant position. The minimum of the *structured non orthogonal* and *unstructured* mesh is lower than the reference value. In addition, the maximum value of $\bar{u}_x$ for these meshes is lower too.

Chapter 6

Conclusions and future considerations

Computational Fluid Dynamics has experienced a growth in the last decades and it seems to be one of the most useful tools in the future. The advantages in time and cost of the simulations are very competitive according to the experimentation. Nevertheless, it is easy to make mistakes during the implementation of the problem and do not perceive it until the discussion of the results, when the money has already been spent. The development of an inappropiate mesh is one of the most common errors, among others. However, there are many methods to improve the mesh quality.

In this Thesis, the aim is to investigate about the effect of a poor mesh quality on the result. Generally, after the generation of the mesh, it is important to check the parameters related to the *mesh quality*, which are often overlooked. We have studied the effect of the *aspect ratio*, *orthogonality* and *skewness* generating four types of mesh for the *Inner ring* (Figure **??**). First of all, we consider the *aspect ratio* insignificant because its location probably does not affect to the results. Secondly, the value of the *orthogonality* obtained for all cases was similar. It affects in the same way to all results, so it is not of interest. Lastly, the *skewness* has a different value for each mesh. As a consequence, we are going to investigate its effects in the solution.

If we move into the figure **??** (d), *structured non orthogonal* mesh reaches the highest values of skewness for a large percentage of elements, followed by the *unstructured* mesh. However, the *structured/unstructured* mesh has a low value of skewness for the most of its elements. Consequently, the solutions of *structured non orthogonal* and *unstructured* cases are not approximate to the results of the reference case, the *structured orthogonal* mesh. On the contrary, the results of the *structured/unstructured* mesh are fairly accurate. After the analysis of different types of results, we observe that the same pattern is met in the solutions of many different variables. Hence, we could conclude that there is a direct relationship between the skewness and the accuracy of the solution.

Despite having reached a logical conclusion with this Thesis, it should go into detail about this topic because of its importance. The symmetry of the *unstructured* mesh could be one aspect to improve in the future to get better results. In addition, the section of the

wake is discretized by a *structured orthogonal* grid. It should be noted that this mesh is the most accurate if it is aligned with the flow. The *wake* is a significant part of the domain because all information about the detachment of the flow behind the cylinder is get in this zone. In order to understand its influence on the final solution, the *wake* grid could be modified, changing again the type of mesh with different qualities, as a future proposal.

Bibliography

[1] H. Jasak. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. Ph.D. for Mechanical Engineering, University of London and Imperial College of Science, Technology and Medicine, 1996.

[2] L. Bruno, D. Fransos, N. Coste and A. Bosco. 3D flow around a rectangular cylinder: A computational study. *Journal of Wind Engineering and Industrial Aerodynamics*, 98: 263-276, 2010.

[3] L. Bruno, Maria V. Salvetti and F. Ricciardelli. Benchmark on the Aerodynamics of a Rectangular 5:1 Cylinder: An overview after the first four years of activity. *Journal of Wind Engineering and Industrial Aerodynamics*, 126: 87-106, 2014.

[4] R. Fattah, D. Angland and X. Zhang. A priori grid quality estimation for high-order finite differencing. *Journal of Computational Physics*, 315: 629-643, 2016.

[5] Y.Cao and T.Tamura. Large-eddy simulations of flow past a square cylinder using structured and unstructured grids. *Computers and Fluids*, 137: 36-54, 2016.

[6] John H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer, Berlin, 3rd edition, 2002.

[7] John D. Anderson Jr. *Fundamentals of Aerodynamics*. McGraw-Hill Series in aeronautical and aerospace engineering, New York, 3rd edition, 2001.

[8] FLUENT 6.3 User's Guide

[9] ANSYS FLUENT 12.0 User's Guide

[10] D. Fransos. *Stochastic Numerical Models for Wind Engineering*. Ph.D. in Mathematics for Engineering Sciences, Politecnico di Torino, 2008.