

AFC applications to reduce the drag coefficient while increasing lift on a NACA airfoil.

Nil Couto Ovejero

Director:
Josep Maria Bergadà Granyó

Bachelor Degree Thesis
Bachelor's degree in Aerospace Vehicle Engineering

School of Industrial, Aerospace and Audiovisual Engineering of
Terrassa
Universitat Politècnica de Catalunya

June 2020

Honor declaration

I declare that,

the work in this Degree Thesis is completely my own work,
no part of this Degree Thesis is taken from other people's work without giving
them credit,
all references have been clearly cited,
I'm authorised to make use of the company's related information I'm providing in
this document.

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

Nil Couto Ovejero

March 24, 2024

Student name

Signature

Date

Title of the Thesis : *AFC applications to reduce the drag coefficient while increasing lift on a NACA airfoil.*

Acknowledgments

I would like to express my sincere gratitude to the director of my Thesis, D.Eng. Josep Maria Bergadà Granyó, for the continuous support, enthusiasm, patience and knowledge. I'm deeply grateful for everything he has taught me.

Moreover, I would like to thank my family for all the support and motivation that they have given to me during the last four months.

Abstract

The aim of this Thesis is to study the influence of active flow control (AFC) in order to improve the aerodynamic efficiency of a 2D NACA airfoil. The studies have been conducted using a computational fluid dynamics (CFD) approach.

In first place, a brief introduction to AFC technologies and CFD has been presented to the reader. The governing equations, turbulence models, typical solvers and AFC typologies have been discussed, and zero-net-mass-flux (ZNMF) jets have been selected as the jet typology used.

A case study definition has been performed, where NACA 8412 airfoil has been established as the airfoil used. In addition, the physical conditions of the problem, non-dimensional parameters used and software utilized have been presented.

The problem resolution has began by writing a MatLab script to automatically generate the structured sub-meshes surrounding the airfoil. Once they have been generated, the unstructured sub-mesh has been designed to fill the domain. The automation of the mesh is intended to simplify the Mesh Independence Test (MIT), which requires a set of meshes to obtain the most computationally efficient.

Previously to the simulation phase, the OpenFoam configuration used must be specified. Apart from the folders used to set the geometrical and flow properties, both the solver and turbulence model have been selected.

The next step has been to carry out the Mesh Independence Test. To do so, 4 mesh candidates with different y^+ values have been simulated. The MIT is aimed to certify that the results are independent of the mesh density. The results for the mesh selected have been considered the baseline case.

When a correct mesh has been obtained, AFC has been introduced based on the results obtained for the baseline case. The velocity signal used in the ZNMF jet has been studied, which includes the forcing frequency and the velocity signal amplitude. In addition, the location of the jet in the airfoil has been evaluated.

Finally, AFC cases have been studied in order to maximize lift and minimize drag. In first place, different values of the jet's inclination angle θ and jet moment coefficients C_μ have been studied to select a first approach of the optimum configuration. With the optimum case, a set of forcing frequencies F_+ have been studied, and the optimum configuration has been evaluated.

Contents

Honor Declaration	i
Acknowledgments	iii
Abstract	v
Contents	vii
List of Figures	xi
List of Tables	xiii
Aim	xv
Scope	xvii
Justification	xix
Requirements	xxi
Nomenclature	xxiv
Acronyms	xxv
List of Abbreviations and Common Terms	xxv
1 Introduction	1
1.1 Computational Fluid Dynamics	1
1.1.1 Boundary Layer	1
1.1.2 Governing Equations	4
1.1.3 Space Discretization Models	5
1.1.4 Flow solver	6
1.1.5 Turbulence Model	7

1.2	Active Flow Control	9
1.2.1	Fluidic actuators	9
1.2.2	Practical applications of AFC	12
2	Case of study	13
2.1	Airfoil selection	13
2.2	Baseline case definition	14
2.3	Non-dimensional parameters used	15
2.4	Software used	17
2.4.1	GMSH	18
2.4.2	MATLAB	18
2.4.3	OpenFoam	18
2.4.4	ParaView	19
2.4.5	Amazon Web Services	19
3	Initial mesh design	21
3.1	Airfoil discretization	21
3.2	Domain geometry	23
3.3	Mesh typologies	24
3.3.1	Structured mesh	24
3.3.2	Unstructured mesh	27
3.4	Mesh design process	28
3.4.1	Structured sub-mesh design	28
3.4.2	Unstructured sub-mesh design	31
3.4.3	Mesh extrusion and export	33
3.4.4	Mesh quality check	34
4	OpenFoam configuration	37
4.1	OpenFoam background	37
4.2	Flow solver selection	38
4.3	Turbulence Model selection	39
4.4	Time directory	40
4.4.1	Velocity file U	40
4.4.2	Pressure file P	42
4.4.3	Turbulent kinematic viscosity file ν_{t}	44
4.4.4	Modified turbulent kinematic viscosity file ν_{Tilda}	45

4.5	Constant directory	47
4.5.1	Transport Properties	47
4.5.2	Turbulence Properties	48
4.5.3	PolyMesh	48
4.6	System directory	51
4.6.1	controlDict	51
4.6.2	fvSchemes	53
4.6.3	fvSolution	54
4.6.4	decomposeParDict	55
5	Mesh independence test	57
5.1	Mesh candidates	57
5.1.1	MIT criteria followed	59
5.2	MIT results	60
5.2.1	Residuals	61
5.2.2	Efficiency comparison	64
5.3	Baseline case selection	66
6	AFC implementation	69
6.1	Jet location	69
6.1.1	Mesh modification	71
6.2	AFC velocity input	73
6.2.1	Forcing frequency	73
6.2.2	Velocity amplitude	74
6.3	OpenFoam set up for AFC	75
7	AFC Results	79
7.1	AFC results	80
7.1.1	θ and C_μ analysis	80
7.1.2	$F+$ analysis	84
7.2	Optimal configuration analysis	87
8	Conclusions	93
8.1	Future work	95
	Bibliography	97

List of Figures

1.1	Boundary layer properties [1]	2
1.2	Turbulent sublayer classification [2]	4
1.3	Flow control actuators [3]	10
1.4	Practical form of a ZNMF actuator [4]	11
2.1	NACA 8412 generated in Chap. 3	14
3.1	NACA 8412	23
3.2	Domain geometry	24
3.3	Non-body fitted mesh approaches [5]	26
3.4	Body fitted mesh approaches [5]	27
3.5	Connection between boundaries of the sub-meshes . . .	29
3.6	Mesh design	31
3.7	Unstructured mesh first approach	32
3.8	Final mesh design	33
5.1	Studied meshes for the MIT	58
5.2	C_l vs time for each mesh candidate	60
5.3	C_d vs time for each mesh candidate	60
5.4	Studied meshes for the MIT	61
5.5	Residuals for $y+=0.7$	62
5.6	Residuals for $y+=0.5$	62
5.7	Residuals for $y+=0.3$	63
5.8	Pressure coefficient comparison	66
5.9	Friction coefficient comparison	66
5.10	Velocity field for the baseline case	68
6.1	C_F of the baseline case	70

6.2	Jet positioning over the airfoil	71
6.3	Mesh for AFC implementation	72
6.4	Scheme of the mesh for AFC implementation	72
6.5	FFT of lift coefficient	74
6.6	Umean graphic representation	75
6.7	Angle between vectors defining scale	78
7.1	C_l comparison for the different θ and C_μ studied	82
7.2	C_d comparison for the different θ and C_μ studied	83
7.3	E comparison for the different θ and C_μ studied	83
7.4	C_l comparison for the F+ studied	85
7.5	C_d comparison for the F+ studied	86
7.6	E comparison for the F+ studied	86
7.7	Velocity Residuals for $F+=4$, $\theta=40^\circ$ and $C_\mu = 0.003$	87
7.8	Pressure Residuals for $F+=4$, $\theta=40^\circ$ and $C_\mu = 0.003$	88
7.9	C_P for $F+=4$, $\theta=40^\circ$ and $C_\mu = 0.003$	89
7.10	C_F for $F+=4$, $\theta=40^\circ$ and $C_\mu = 0.003$	89
7.11	C_l VS time for the optimum configuration and the baseline case	90
7.12	C_d VS time for the optimum configuration and the baseline case	90
7.13	Velocity field comparison	91
7.14	Pressure contour comparison	91

List of Tables

2.1	Physical conditions	15
5.1	Mesh quality comparison	58
5.2	Mesh Independence Test	64
7.1	AFC results comparison	81
7.2	F+ comparison	84

Aim

The aim of this Thesis is to improve the aerodynamic efficiency of the NACA 8412 airfoil by using a zero-net-mass-flux jet. The study of the jet influence on lift, drag, pressure and shear stresses has been performed with OpenFoam, a CFD open-source software.

The objective of the simulations is to observe how the aerodynamic coefficients respond to different jet momentum coefficients, forcing frequencies and jet inclination angles. By modifying this parameters, the most efficient jet configuration which maximizes lift and minimizes drag is intended to be found.

Scope

This Thesis will consist of implementing Active Flow Control in a 2D NACA airfoil using a CFD based resolution. The project will be completed by June, 2020.

Active Flow Control technologies will be introduced to the reader, with the objective of presenting its benefits and the main jet typologies used. However, other jet typologies' theoretical background and implementation will not be described. The design of the synthetic jet mechanism has been considered out of scope, focusing the project on its aerodynamic optimization. A introduction to Computational Fluid Dynamics will be presented too.

The case of study will be defined, including the airfoil used, baseline case properties, software used and non-dimensional parameters present in the project.

The OpenFoam configuration will be presented, containing a brief introduction to the software, flow solver discussion and turbulence model selection. In addition, the set up of all the folders included in the simulation will be presented and described. Both solver and turbulence model validation have been considered out of the scope.

The Mesh design will include a MatLab script in order to automatize the mesh generation in GMSH. A brief description of different meshing approaches will be included. A study of the mesh independence will be developed to ensure the results reliability, and the baseline case's mesh will be selected.

Active Flow Control will be implemented to the baseline case. The implementation will include a justification of the input velocity signal and of the location the slot.

Finally, simulations including Active Flow Control will be performed, and the results obtained will be compared with the baseline case in order to maximize

aerodynamic efficiency. The evaluation of the results will be performed using the time averaged lift and drag coefficients, as well as pressure and wall shear stress distribution along the chord. In addition, ParaView will be used to visualize the flow around the airfoil. Even though the jet position and its width influence on AFC performance, the modification of these values have been considered out of scope.

The optimal jet configuration will be presented and conclusions will be extracted.

Justification

It's well known that the aerospace industry is facing a major challenge: the reduction of CO_2 emissions and a transition into a more sustainable industry model. Given the high ecological impact of this industry, the development of more eco-friendly technologies is a fundamental step in the aviation history, ensuring an ethical model for the decades to come. Active flow control could be a feasible solution to this problematic, as it shows fuel consumption improvements that exceeds any of its precedents. According to [6], for transport-type airplanes, e.g., the fuel burned might be decreased by the 30 percent. In addition, decreasing the fuel required to operate an aircraft would reduce the operation costs of commercial aviation, reducing the ticket prices for air travel.

AFC does not only affect drag reduction, and consequently fuel consumption, but also stability, maneuverability, aircraft endurance or noise reduction. For example, according to [6], active flow control technologies would offer the capability of operating transport airplanes that can fly without refueling for approximately 24 hours. Furthermore, AFC could be used as a substitute for much heavier high lift devices such as flaps or slats, if implemented for take-off and landing situations.

AFC technologies are still in a development phase too premature to be implemented in commercial aviation. Nonetheless, the revolutionary improvements in fuel consumption, maneuverability or noise reduction are the reason why this technology has been studied worldwide during the last decades. For this reason, the present Thesis could be used in the future by other researchers in order to implement this technology in experimental studies.

Requirements

The Thesis must be finished within four months, and submitted complying with the guidelines by the university.

The basic requirement of the Thesis is to implement AFC in order to improve the aerodynamic efficiency of a 2D NACA airfoil. To do so, the drag must be reduced and the lift increased in comparison with the baseline case.

The case of study must be selected, specifying the Reynold number, angle of attack, free-stream velocity, etc.

The computational domain must be meshed in order to simulate the studied cases. In order to do so, the software GMSH is required. The sensitivity of the mesh must be studied in order to observe if the results are dependant on the mesh density. This study will help to evaluate which mesh is the most efficient one in computational terms.

The solver used must be evaluated using bibliographical references. The turbulence model used in the CFD studies must be decided too. Neither the solver nor the turbulence model will be validated, as it's been considered out of scope.

Jet moment coefficient, forcing frequency and jet inclination angles have to be studied in order to obtain the most efficient configuration.

Nomenclature

δ	Boundary layer thickness [m]
δ_T	Turbulent boundary layer thickness [m]
μ	Dynamic viscosity [kg/(m s)]
ν	Kinematic viscosity [m ² /s]
ρ	Freestream density [kg/m ³]
τ_w	Wall shear stress [Pa]
f_0	Vortex shedding frequency [Hz]
θ	Jet inclination angle [°]
$\tilde{\nu}_t$	Modified turbulent kinematic viscosity [m ² /s]
C	Chord length [m]
C_μ	Jet momentum coefficient
C_d	Drag coefficient
C_F	Friction Coefficient
C_l	Lift coefficient
C_P	Pressure coefficient
E	Aerodynamic efficiency
F^+	Non-dimensional forcing frequency
F_d	Dimensional drag force [N]
F_l	Dimensional lift force [N]
h	Jet width [m]
L	Characteristic length [m]
Ma	Mach number
P_∞	Freestream static pressure [Pa]
Re	Reynolds number
u^+	Dimensionless velocity
U_∞	Freestream velocity [m/s]

u_τ Friction velocity [m/s]

U_{max} Maximum jet velocity [m/s]

y^+ Dimensionless wall distance

Acronyms

AoA Angle of attack

AFC Active flow control

AWS Amazon Web Services

CFD Computational fluid dynamics

CFL Courant-Friedrichs-Levy number

EC2 Elastic Compute Cloud

FDM Finite Differences Method

FVM Finite Volume Method

FEM Finite Element Method

HLD High lift devices

ISA International Standard Atmosphere

LE Leading edge

LES Large Eddy Simulation

MIT Mesh Independence Test

NMF Nonzero mass flux

RANS Reynolds Averaged Navier-Stokes

TE Trailing edge

ZNMF Zero net mass flux

Chapter 1

Introduction

The objective of this chapter is to introduce Active Flow Control (AFC) technologies to the reader, as well as briefly describe the jet typology chosen for the following studies. In addition, computational fluid dynamics (CFD) has been presented to the reader.

1.1 Computational Fluid Dynamics

1.1.1 Boundary Layer

In order to understand how AFC influences on the aerodynamic efficiency of an airfoil, it's necessary to introduce the concept of boundary layer and its main characteristics.

According to [1], the boundary layer is the thin region of flow adjacent to a surface, where the flow is retarded by the influence of friction between a solid surface and the fluid. The non-slip condition states that the flow velocity at a given surface is zero. As a result, the flow velocity increases progressively in the y direction until its value it's equal to the freestream velocity.

The boundary layer thickness δ corresponds to the distance for which the velocity is the 99% of the freestream velocity. A similar approach can be done for temperature, which must match the surface temperature at $y=0$. δ_T is the thickness of the

boundary layer for which the temperature is the 99% of the freestream temperature. For any given x value, both temperature and velocity change as function of the distance y . Thus, for any x value, a different velocity and temperature profile can be observed, as it can be seen in Fig. 1.1. Even though two boundary layers can be observed in Fig. 1.1 (velocity and temperature boundary layer), the following explanation will be focused on the velocity boundary layer, as temperature is not treated in the present Thesis.

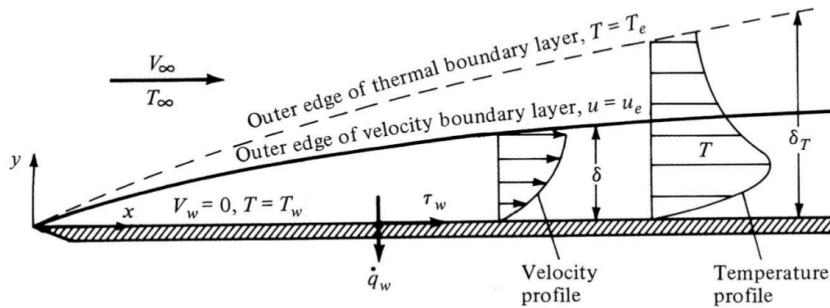


FIGURE (1.1) Boundary layer properties [1]

Boundary layers can be classified in two typologies: laminar and turbulent. Non-dimensional parameter Reynolds number (Re) takes an important role when defining boundary layer typologies, since it associates inertial and friction forces. It can be expressed as:

$$Re = \frac{LU_\infty}{\nu} = \frac{\rho LU_\infty}{\mu} \quad (1.1)$$

Where L is a characteristic length, ρ is the fluid density, U_∞ the freestream velocity, ν the kinematic viscosity of the fluid and μ the dynamic viscosity.

It's considered that the transition from laminar to turbulent takes places at $Re = 10^5$. For incompressible laminar flow over a flat plate, the boundary layer thickness can be expressed as follows:

$$\delta = \frac{5.0x}{\sqrt{Re_x}} \quad (1.2)$$

For incompressible turbulent flow over a flat plate, the boundary layer thickness can be expressed as:

$$\delta = \frac{0.37x}{Re_x^{1/5}} \quad (1.3)$$

By substituting Eq.(1.1) in Eq.(1.3) it can be observed how for the turbulent boundary layer $\delta \propto x^{4/5}$. On the other hand, doing an analogue substitution for laminar boundary layer δ presented in Eq.(1.2), it's observed how $\delta \propto x^{1/2}$. As a result, it can be stated that the boundary layer thickness grows faster with distance for turbulent boundary layers.

Turbulent boundary layers are governed by randomness, requiring of turbulence models in order to predict how the flow properties change with time. A brief introduction to turbulence modeling has been presented in Sec. 1.1.5. After the transition, the turbulent boundary layer can be classified in multiple sublayers. This sublayers depend on the near-wall region where the flow is studied, and are classified according to the dimensionless wall distance y^+ . It is a non-dimensional distance used to evaluate the relative influence of viscous and turbulent processes, and can be defined as follows:

$$y^+ = \frac{u_\tau y}{\nu} \quad (1.4)$$

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \quad (1.5)$$

Where u_τ is the friction velocity and τ_w the wall shear stresses. Previously to the sublayers description, dimensionless velocity must be introduced. This parameter denotes the ratio between the velocity u parallel to the wall as a function of y and the friction velocity u_τ , as presented in Eq.(1.6).

$$u^+ = \frac{u}{u_\tau} \quad (1.6)$$

The shape of the velocity profile is related to u^+ , since it's a function of y^+ . Depending on the zone of the near-wall region where the flow is studied, three approaches for the dimensionless velocity can be followed [7] [2]:

- **Viscous sub-layer, ($y^+ < 5$)**

It's a thin layer next to the surface where the flow is governed by viscous effects. Even though the flow is not strictly laminar (it experiences random fluctuations in velocity), the transport of fluid momentum is dominant over turbulent transport of momentum. Subsequently, shear stresses can be assumed equal to the wall shear stress τ_w , and the dimensionless velocity u^+ can be expressed as $u^+ = y^+$.

- **Buffer layer or blending region, ($5 < y^+ < 30$)**

In this region, located outside the viscous sub-layer, the flow may be considered turbulent, but still not fully developed. As a result, both viscous shear stress and turbulent shear stress must be contemplated, being commonly understood as a transient zone in shear stress terms. Moreover, there's not an expression for the dimensionless velocity u^+ .

- **Fully turbulent or log-law region, ($y^+ > 30$ to 60)** In this region the flow it's completely turbulent, being governed by the turbulent shear stress. This stress suffers very low modifications with the distance to the wall, and a logarithmic relation between non-dimensional velocity and non-dimensional distance can be established.

$$u^+ = \frac{1}{k} \ln(y^+) + B \quad (1.7)$$

Where K refers to the Von Karman's constant, B is related to wall roughness.

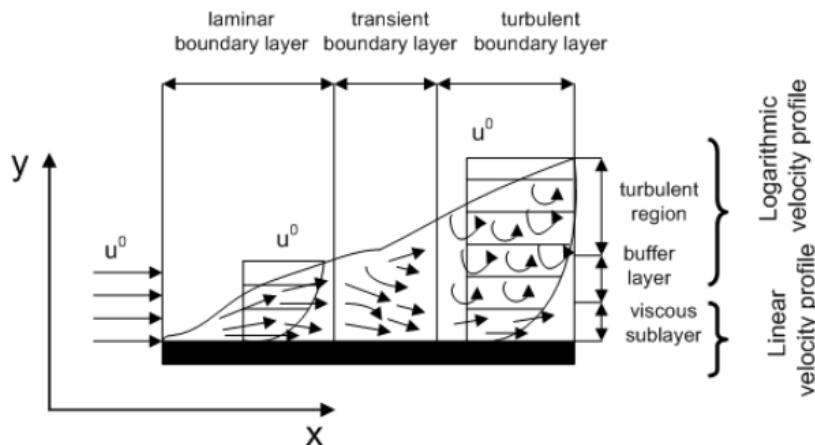


FIGURE (1.2) Turbulent sublayer classification [2]

1.1.2 Governing Equations

CFD is based on the resolution, using approximations, of the Navier Stokes Equations. According to Chap. 2, the 2D studied cases must be treated as unsteady, turbulent and incompressible. Thus, the governing equations of the problem are

the Navier Stokes equations for incompressible two-dimensional flow. These equations are presented in Eq.(1.8), Eq.(1.9) and Eq.(1.10).

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} \frac{\partial(\rho v)}{\partial y} = 0 \quad (1.8)$$

$$\rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) = - \frac{\partial p}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \rho g_x \quad (1.9)$$

$$\rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) = - \frac{\partial p}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + \rho g_y \quad (1.10)$$

1.1.3 Space Discretization Models

CFD needs to solve the differential equations previously presented in discrete points of the domain (mesh points). Thus, discretization models are required to organize the domain in points where the solution will be obtained, and to transform the differential equations in algebraic expressions. The main discretization models used in CFD are presented below.

Finite Differences Method

Finite Differences Methods (FDM) are based in approximating the differential terms by replacing the partial derivatives in the equation with differential quotients [8]. Taylor series are usually used as algebraic difference quotients, and the accuracy of the result is related to the terms of the series (the more terms, the more accurate the solution). The main idea behind FDM is related to the derivative's definition of a smooth function u at a point x , presented in (1.11)

$$u'(x) = \lim_{h \rightarrow 0} \frac{u(x + h) - u(x)}{h} \quad (1.11)$$

When h tends to 0, its value is sufficiently small to provide a good approximation of the derivative. However, as the method is based on approximations, there's an error between the exact solution and the computed solution. This error is known as discretization error or truncation error, and represents the neglected terms in

the Taylor series. As a result, an approximation can be considered correct if the truncation error tends to 0 when h tends to 0.

Finite Volume Method

Finite Volume Method (FVM) is a control volume based discretization technique extensively used in CFD. The domain is divided into control volumes, and the variable to be computed is numerically located on the centroid of the volume. The differential form of the governing equations are integrated for each control volume. The variation of the variable between volumes is described using interpolation profiles. The resulting equation is known as the discretized equation. This discretization technique is highly recommended with discontinuous solutions due to its conservative nature [9].

Finite Element Method

Finite Element Method (FEM) is a numerical method used to solve differential equations using finite elements. This technique discretizes the domain into elements, and computes the solution in the nodes generated. As a result, the accuracy of the results is influenced by the number of elements placed in the domain. FEM is based on weighted residuals methods [10], which means that the solution is approximated by a finite sum of weight functions. As a result, the function coefficients obtained are intended to minimize the error between the computed solution and the exact solution.

1.1.4 Flow solver

Three solvers usually used in AFC related problem have been presented to the reader: *SimpleFoam*, *PisoFoam* and *PimpleFoam*.

SimpleFoam is an incompressible steady-state solver which uses the SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm [11]. It can solve both laminar and turbulent cases, requiring variables characterizing turbulence for the latest case.

PisoFoam is a transient solver for incompressible, turbulent flows that follow the PISO algorithm (Pressure-implicit with Splitting of Operators) [12]. It's an efficient approach to solve RANS equations, which means that the time-dependent Navier-Stokes equations are computed capturing the transient behaviour with relatively small computational resources.

PimpleFoam is a solver for incompressible, unsteady turbulent flows. The algorithm followed is a combinations of PISO and SIMPLE algorithms. The combination of both results on a resolution similar to the followed by the PISO algorithm. Despite that, PIMPLE algorithm includes outer correctors, which improves the stability of the resolution for high time-step simulations where the maximum Courant number must be above 1, or when the solution is unstable by nature.

1.1.5 Turbulence Model

According to [5], two families of turbulence models can be considered: Large Eddy Simulation (LES) based models and Reynolds Averaged Navier-Stokes (RANS) based model.

LES-based models perform a direct computation of the turbulent flow fluctuations in space and time above a certain length scale. Below this scale (subgrid scale), the solution is obtained using semi-empirical laws.

The RANS-based turbulence models obtain the solution by calculating the turbulent averaged flow and its averaged fluctuations. As a result, RANS model approximates the turbulent flow by modelling the stresses, and computing larger scales than in the LES model. Due to its computational efficiency, RANS-based models are the most widely used in CFD, and has been selected as the turbulence model family used for the present Thesis.

Regarding RANS-based models, different methodologies can be followed in order to solve the averaged Navier-Stokes equations. According to [13], the main turbulence models for solving RANS equations are:

- **Spalart-Allmaras** : This low Re model adds an additional variable (modified turbulent kinematic viscosity $\tilde{\nu}_t$) to solve the transport equation for

eddy turbulent viscosity. It was created with aerodynamics purposes, and can solve the complete domain down to the solid surfaces. It shows a great robustness in the resolution and easily converges if the Courant number is below the required for the given case.

- **L-VEL and algebraic yPlus** : These turbulence models are based on algebraic expressions focusing on the local fluid velocity and distance to the wall in order to compute the eddy viscosity. They show good approximations for internal flows, and demand relatively low computational power. Nonetheless, they are the less accurate turbulence models among the presented in this subsection.
- **$k-\varepsilon$** : The $k-\varepsilon$ is a two equation model which computes the solution by using two transport equations. The two variables added to the system are: turbulence kinetic energy k and the rate of dissipation of turbulence kinetic energy ε . It has been observed that it's a suitable model for free-shear flows, for example for relatively small pressure gradients. However it shows a relatively high lost of accuracy when treating large adverse pressure gradients, strong curvature or jet flows [14].
- **$k-\omega$** : This model, as in the $k-\varepsilon$ case, is a two equation model to solve turbulent flows. The first turbulence parameter introduced is the turbulence kinetic energy k , and the second parameter is the specific rate of dissipation of kinetic energy ω . This model is suitable for low Re cases, as well as with wall functions. The convergence of the result is more difficult than in the $k-\varepsilon$ due to its non-linearity. It's a useful model for internal flows, flow around a strong curvature or separated flows.
- **SST**: The SST model, or Menter's Shear Stress Transport model, is a combination of the two last models described. It's a low Re model, and has similar computational requirements as the $k-\omega$, even though its formulation is slightly different.
- **Low Re $k-\varepsilon$** : As its name suggests, this model is similar to the $k-\varepsilon$ previously described, with the main difference that this model does not need wall functions. It requires more dense meshes than the $k-\varepsilon$ for the whole domain, and a good initial estimation is required in order to successfully converge

the solution. This model has shown to compute lift and drag with greater accuracy than the $k-\varepsilon$ model, a greater capacity to predict separation and reattachment in certain low Re cases.

- **v2-f** : The $v2-f$ model is a four-equation model, two of them previously explained in the $k-\varepsilon$ model: turbulence kinetic energy (k) and dissipation rate (ε). The first equation remaining describes the transport of turbulent velocity fluctuations normal to the streamlines, and the second one is related to the redistribution of kinetic turbulence energy between the normal and parallel directions. It has shown accurate results for confined flows around a surface.

1.2 Active Flow Control

Flow control technologies, whereas they act actively or passively, have the objective to modify the flow behaviour around a given surface. The objective of these modifications is to improve the aerodynamic efficiency, control the transition from laminar to turbulent, prevent/induce flow separation, enhance/eliminate turbulence, suppress flow-induced noise, etc. [15]. Even though many passive flow control technologies have been successfully applied (ribblets, fences, vortex generators, etc), the aeronautical industry has had a growing interest on AFC during the last decades, as it allows to be activated under certain situations and in different parts of the airplane independently.

1.2.1 Fluidic actuators

AFC systems can be classified according to its actuator, which can be fluidic, mechanical, plasma based or electromagnetic, among others [15] (see Fig. 1.3). However, as the present Thesis focuses on fluidic actuators, only this typology will be described. These actuators are the most common ones, and have been extensively studied during the recent decades.

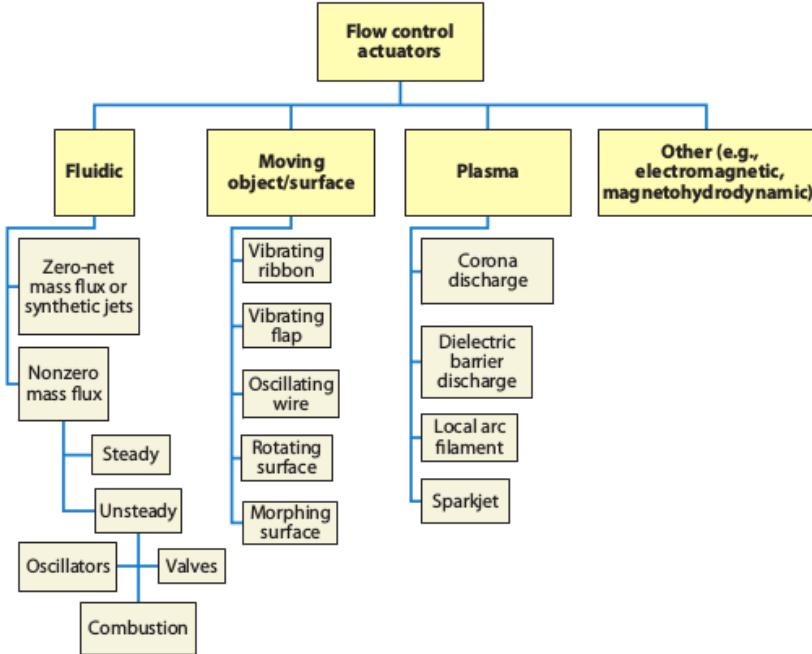


FIGURE (1.3) Flow control actuators [3]

One of the main advantages of fluidic actuators is the lack of moving parts in contact with the external flow. As shown in Fig. 1.3, fluidic actuators can be classified according Zero-net-mass-flux (ZNMF) or synthetic jets and Nonzero-mass-flux (NMF). NMF jets can be steady, which means that they only blow or suction fluid, or unsteady, blowing and suctioning fluid in different phases. The main difference between ZNMF and unsteady NMF jets is that ZNMF periodically alternates between an air suction phase and an air ejection phase, following a sinusoidal input velocity signal. In contrast, unsteady NMF don't alternate periodically, having different phase length for the suction and ejection phases. As a result, ZNMF jets do not require any fluid source due to the acquisition of fluid flow from the external flow. Since the the same ammount of fluid is used for both phases, the mass balance at the jet's orifice is zero. On the other hand, Nonzero mass flux jets require from a fluid source in order to blow or suction fluid.

According to [16], periodic forcing actuators show greater improvements on aerodynamic efficiency and separation delay in respect with continuous jets. For this reason, the present Thesis will focus on the study of periodic forcing jets, more specifically ZNMF jets.

ZNMF jets can efficiently reattach the boundary layer by periodically blowing or suctioning fluid into the external flow. According to [16], during the suction phase the low momentum fluid of the boundary layer is forced to enter the jet's cavity, and the high momentum fluid tends to move toward the boundary layer edge nearer to the surface. On the other hand, during the blowing phase, high momentum fluid is expelled from the jet's cavity into the inner zones of the boundary layer. Moreover, during the actuation cycle two counter-rotating vortices are generated (each one as a consequence of one of the actuation cycles), which enhances the mixing process in the closest zone to the surface of the boundary layer. Consequently, the combination of both phases result in a partial replenish of the momentum deficit in the boundary layer, enhancing the delay of the flow separation and the mitigation of the adverse pressure gradient. Even though it has been previously stated that the mass balance at the jet's orifice is zero, it must be noted that the momentum balance is not zero, implying an energy transfer from the jet to the external flow.

The most frequent practical form of a ZNMF actuator consists of a piezoelectric disk attached to a metal diaphragm to form like a sealed cavity [4]. The oscillation membrane is placed in the bottom of the cavity, and opposite to it there is the jet slot. The membrane follows a periodic oscillation, forcing the flow to periodically enter and evacuate the cavity through the jet orifice. A graphic representation of ZNMF actuators' structure is presented in Fig. 1.4. The jet mechanism won't be designed, as the present Thesis focuses on the aerodynamic improvement using ZNMF technology.

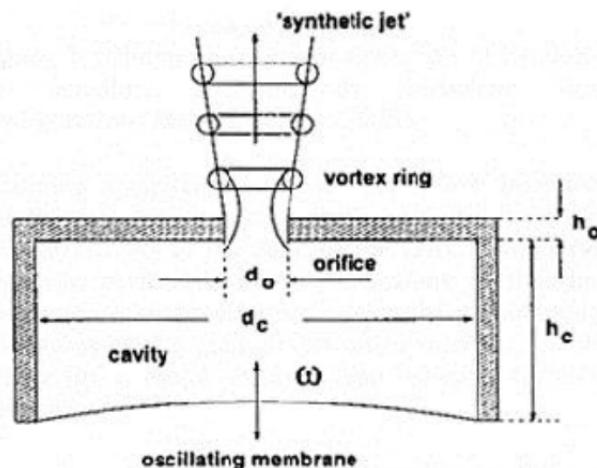


FIGURE (1.4) Practical form of a ZNMF actuator [4]

1.2.2 Practical applications of AFC

The main contributions on AFC still need more maturity in order to be feasibly applied in the aeronautical industry [17]. However, the benefits in fuel consumption, stability, maneuverability and aircraft endurance that it would suppose makes it one of the main options to transition into a more sustainable and secure industry model. In addition, AFC could be useful to replace some existing components of the aircraft (e.g high lift devices), reducing the weight of the aircraft and eliminating critical moving parts.

On civil aviation, AFC could be used in multiple components of the aircraft, as smooth surfaces like wings or fuselage, on after-bodies or on junctions. Wings are the main source of lift of the aircraft but, as a consequence, they generate a great percentage of induced drag. For this reason, wings are optimized for the cruising flight, requiring high-lift devices (HLD) to ensure a correct aircraft's performance on other stages of the flight like take-off or landing. These devices allow to modify the wing's surface and, therefore, modify the pressure distribution to generate extra lift when the wing is under high angles of attack. However, even if HLD are implemented, some regions of the wing like the wing root, wing-pylon junction and the wingtip are susceptible of suffering airflow separation. Some examples of how AFC could be applied to overcome airflow separation in some components of the aircraft are presented on the paragraph below.

The actual aeronautical industry tendency regarding power plants is to use high/ultra-high bypass engines. For this reason the engine nacelle must be located closer to the wing to ensure a secure operation of the aircraft while doing ground operation. As a result, leading edge slats cannot be used in the wing-pylon junction. During take-off and landing operations this junction is dominated by seven different types of vortex , whose incidence could be minimized by the usage of AFC [18]. Another example can be found in winglets, which are designed to reduce tip vortex induced drag during cruise flight. However, when high angles of attack are present, this surfaces cause a lost of total lift. Nevertheless, according [19], an offset stall by 2.4° , increase of maximum lift by more than 10% and a reduction of drag by 37% was achieved using AFC with take-off conditions.

Chapter 2

Case of study

The aim of this chapter is to define the geometry to be studied as well as the properties of the flow for the present case. The non-dimensional parameters and the software used have been introduced as well.

2.1 Airfoil selection

The airfoil studied in the present Thesis is the NACA 8412, which belongs to the NACA 4-digit series family. The geometry of the airfoil defines the numbering system for this series. The first integer indicates the maximum camber in percent of the chord, the second number indicates where the maximum camber is in respect to the leading edge (LE) in tenths of the chord, and the last two integers define the airfoil thickness in percent of the chord.

NACA 4-digit airfoils present relatively low maximum lift values and elevated drag. Thus, it will help to visualize the efficiency improvement using AFC, as the objective is to reduce drag while increasing lift. In addition, airfoils with relatively low thickness ratios (between 10 and 16 percent of the chord length) present a stalling phenomenon called leading-edge stall. This phenomenon is characterized by the abrupt flow separation on the extrados near the LE. As a result, the NACA 8412 airfoil will follow this behaviour, making it simpler to observe the impact of AFC along the top surface of the airfoil [1]. A graphic representation of NACA 8412 airfoil is presented in [3.1](#)

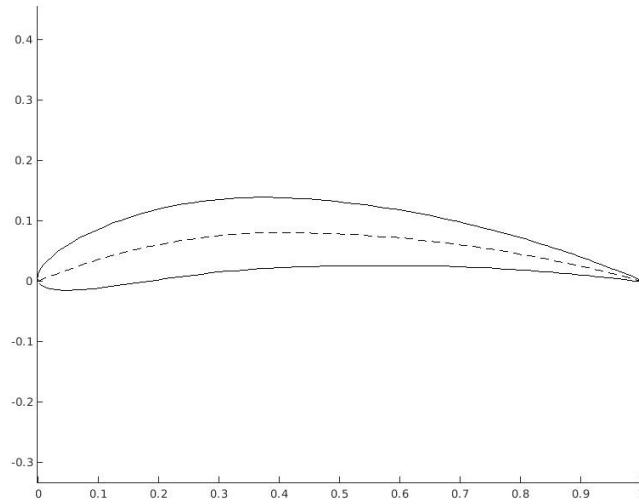


FIGURE (2.1) NACA 8412 generated in Chap. 3

2.2 Baseline case definition

In order to correctly compare the efficiency of AFC application with the baseline case, the same flow properties must be applied for all the simulations. The International Standard Atmosphere (ISA) model at sea level has been used to select the air density and dynamic viscosity. As the Thesis results will be obtained using a numerical method, a low Reynolds number must be selected in order to have relatively low computational requirements and computational time. In addition, due to the academic purpose of the Thesis, the selected Re doesn't have to represent a real aeronautical case, which are characterized by very high Re . The physical conditions used for all the simulations in the present Thesis are summarized in Tab. 2.1.

Reynolds number (Re)	68.5×10^3
Freestream velocity (U_∞)	1 m/s
Kinematic Viscosity (ν)	$1.46 \times 10^{-5} \text{ m}^2/\text{s}$
Density (ρ)	1.225 kg/m^3
Angle of attack (α)	15°
Chord length (C)	1 m

TABLE (2.1) Physical conditions.

2.3 Non-dimensional parameters used

A brief description of each non dimensional parameter used in the present Thesis is presented below.

Lift coefficient

$$C_l = \frac{2F_l}{\rho C U_\infty^2} \quad (2.1)$$

Where C is the chord length, ρ the free-stream fluid density, U_∞ the freestream velocity and F_l is the dimensional lift force. C_l is used to relate the lifting force generated by a body to the fluid density, a reference surface (as the problem approached is 2D, the reference is the chord) and the free-stream velocity. It's a function of the angle of attack (AoA) and Re .

Drag coefficient

$$C_d = \frac{2F_d}{\rho C U_\infty^2} \quad (2.2)$$

Where F_d is the dimensional drag force. C_d is used to quantify the air resistance (also known as drag) over a given object. It's a function of the AoA and Re . As the problem treated is incompressible ($Ma < 0.3$), it can be considered independent of the Mach number.

Aerodynamic efficiency

$$E = \frac{C_l}{C_d} \quad (2.3)$$

Aerodynamic efficiency E , also known as lift-to-drag ratio, it's the quotient between lift and drag (or their respective coefficients). As a result, the usual aerodynamic approach is to maximize the efficiency by maximizing lift and minimizing drag.

Pressure coefficient

$$C_P = \frac{P - P_\infty}{\frac{1}{2}\rho U_\infty^2} \quad (2.4)$$

Where P_∞ is the free-stream static pressure. C_P denotes the ratio between manometric pressure and dynamic pressure. Every point in a flow field has a particular C_P . When $C_P=0$, the pressure is equal to P_∞ , and when $C_P=1$ it indicates a stagnation point.

Friction coefficient

$$C_F = \frac{\tau_w}{\frac{1}{2}\rho U_\infty^2} \quad (2.5)$$

Where τ_w is the wall shear stress. When a fluid flows around a given surface, frictional forces also known as skin friction drag appear to the surface. Different approaches for C_F must be considered whether the flow is laminar (Blasius solution), transitional or turbulent (Prandtl's one-seventh-power law).

Jet momentum coefficient

$$C_\mu = \frac{h(\rho U_{max}^2) \sin \theta_j}{C(\rho U_\infty^2)} \quad (2.6)$$

Where h is the jet width, U_{max} the maximum jet velocity and θ_j the jet inclination angle with respect to the airfoil surface. This non-dimensional parameter will be evaluated in Chap. 6.

Non-dimensional forcing frequency

$$F^+ = \frac{f}{f_0} \quad (2.7)$$

Where f is an arbitrary frequency and f_0 is the vortex shedding frequency. This parameter will be studied in Chap. 6.

Reynolds number

$$Re = \frac{U_\infty C}{\nu} \quad (2.8)$$

Where ν is the kinematic viscosity of the fluid. Re denotes the ratio of inertial forces and viscous forces within a fluid. As a result, it represents whether if the flow is dominated by viscous forces and must be treated as laminar (low Re), or if the flow is turbulent being dominated by inertial forces (high Re).

Courant-Friedrichs-Levy number

$$CFL = \frac{u \Delta t}{\Delta x} \quad (2.9)$$

Where Δt is the time step and Δx the length interval. CFL is a key parameter to ensure the convergence of the results while solving partial differential equations. As a consequence, CFL is used to choose a time step that complies with the maximum CFL permitted (usually, $CFL_{max} = 1$ for explicit solvers). To exemplify its meaning, imagine a wave moving across a grid which amplitude must be computed using discrete time steps. Thus, the time step should be smaller than the time for the wave to travel to the next grid point. Therefore, CFL is used as an indicator of the robustness of a simulation.

2.4 Software used

During the different stages of the Thesis, a variety of software has been used to generate the mesh, simulate the flow and analyzing the results obtained. The

operational system where the software have been installed is Ubuntu 16.04, a Linux based operative system.

2.4.1 GMSH

GMSH is an open source 3D finite-element mesh generator with a built-in CAD engine and post-processor [20]. It contains four modules: geometry, mesh, solver and post-processing. For each module, the input parameters can be introduced using the graphical user interface, using GMSH's script language (.geo files), or using the Application Programming Interface (API), which supports C++, C and Python syntax.

GMSH has been used to design certain parts of the mesh (some parts have been automatically generated using a MatLab script to write .geo files), and to visualize the final mesh.

2.4.2 MATLAB

MATLAB is a numerical computing environment, which uses its own programming language (M). It allows to perform matrix manipulations, algorithm implementation, create user interfaces and plotting data. In addition, it offers a great amount of packages to perform numerical computations of specific engineering branches. MATLAB has consolidated as one of the most widely used programming software in academic environments, such as education or investigation.

It has been used to automatize the mesh generation and to write the mesh parameters in .geo files, in order to be read by GMSH. In addition, during the post-processing stages, it has been used to generate the results plot, due to the limited capabilities of the visualization software ParaView in terms of data plotting.

2.4.3 OpenFoam

OpenFoam is an open-source C++ toolbox used to perform CFD related simulations. It's composed of a set of libraries including solvers and utilities. The

solvers available enables the user to perform calculations with different turbulence models, chemical reaction models, compressible and incompressible flow models, heat transport models, etc. OpenFoam is one of the most popular CFD tools, and has been extensively used worldwide by the industry and the academia due to its reputation and features.

Due to the importance of OpenFoam in this Thesis, a further explanation has been developed in Chap. 4 Sec. 4.1, which includes the definition of the files composing OpenFoam. This software has been used to perform all the simulations included in the present Thesis in order to obtain the velocity fields, pressure fields and the coefficients previously presented (C_l , C_d , C_P and C_F) of each studied case.

2.4.4 ParaView

ParaView is an open-source, multi-platform data analysis and visualization application [21]. It allows to interactively visualize large datasets like the flow solutions obtained using OpenFoam. In addition, it includes a set of functions to visualize different flow properties and plot the results, using a vertical architecture (the functions can be chained).

It has been used to visualize the flow solution obtained from the OpenFoam simulations, as well as to generate some images or videos showing the velocity fields and the pressure contours. Moreover, it has been used to extract the instant values of those parameters that require a time averaged representation.

2.4.5 Amazon Web Services

Amazon Web Services (AWS) is an Amazon service that provides on-demand cloud computing platforms. Although it includes a set of services, only the Amazon Elastic Compute Cloud (EC2) has been used during the present Thesis. EC2 allows the user to use a set virtual cluster computers, giving the freedom to select the CPUs units, GPUs units, local/RAM memory, storage size, and pre-loaded applications. AWS offers fixed prices as function of the computational features selected, and the prize vary if the service is used with spot instances or reserved instances.

AWS has been used to run the OpenFoam simulations included in this project using CFD Direct From the Cloud platform. This platform allows the user to run OpenFoam simulations via a remote desktop using EC2 servers. The usage of AWS was chosen because the computational requirements of the simulations were too expensive to be supported in a personal computer. Spot instances has been used as their prize is considerably lower, but they are susceptible to interruptions if the user's maximum instance price exceeds the current Spot price for a given instance.

The usage of AWS enabled to perform computations in parallel as well as decomposing the domain in a larger number of cores. It must be taken into account that AWS was used in the AFC related simulations. Part of the Mesh Independence Test cases were simulated using the personal computer, which showed that the computational requirements had to be improved in order to complete the later simulations.

Chapter 3

Initial mesh design

The aim of this chapter is to automatize the mesh generation as well as design the initial mesh to be tested in Chap. 5. Even though different meshes will be tested in the mentioned chapter, the methodology followed to design each mesh is the same.

Previously to the mesh design, the airfoil has been discretized, and the studied domain defined. Once the airfoil is represented inside the computational domain, a first structured sub-mesh surrounding the airfoil has been designed in order to capture the variations of the boundary layer. In addition, a second structured sub-mesh around the previously mentioned has been generated to add resolution to the global mesh. Finally, an unstructured mesh has been generated to complete the studied domain as well as a structured mesh to capture the wake rear the airfoil.

3.1 Airfoil discretization

The studied airfoil is the NACA 8412, as it has been previously explained in Chap. 2 Sec. 2.1. However, this airfoil must be represented using points in order to numerically define it. An explanation of the process followed is presented below. The equations used to calculate both the camber lines $y_c(x)$ and thickness distributions $y_t(x)$ of a NACA 4-digit airfoil are:

$$y_t = \pm \frac{t}{0.2} (0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4) \quad (3.1)$$

$$y_c = \frac{f}{p^2} (2px - x^2) \quad 0 \leq x \leq p \quad (3.2)$$

$$y_c = \frac{f}{1-p^2} (1 - 2p + 2px - x^2) \quad p \leq x \leq 1 \quad (3.3)$$

Where t is the thickness in percent of chord, p the maximum camber position in tenths of chord and f the maximum camber in percent of chord, according to the NACA 4-digit numbering system [22]. With the equations defining the camber and thickness distribution, the points located at extrados (x_u, y_u) and intrados (x_l, y_l) can be calculated using:

$$x_u = x - y_t(x) \sin \theta \quad (3.4)$$

$$y_u = y_c(x) + z_t(x) \cos \theta \quad (3.5)$$

$$x_l = x + y_t(x) \sin \theta \quad (3.6)$$

$$y_l = y_c(x) - z_t(x) \cos \theta \quad (3.7)$$

$$\theta = \tan^{-1} \left(\frac{dy_c}{dx} \right) \quad (3.8)$$

Taking into account the expressions above, a MATLAB script (see Appendix ??) has been written to automatically discretize the airfoil with n points for both intrados and extrados zones, as well as drawing the splines along the points. Moreover, a cosine distribution has been used in order to increase the number of points on the leading and trailing edge, where the geometry requires more precision. The number of points defining the airfoil have been set at 200 points, equally distributed between extrados and intrados. Once the points defining the airfoil have been calculated, splines along the points must be generated in order to define the airfoil geometry. The NACA 8412 geometry obtained using the script is presented in Fig. 3.1

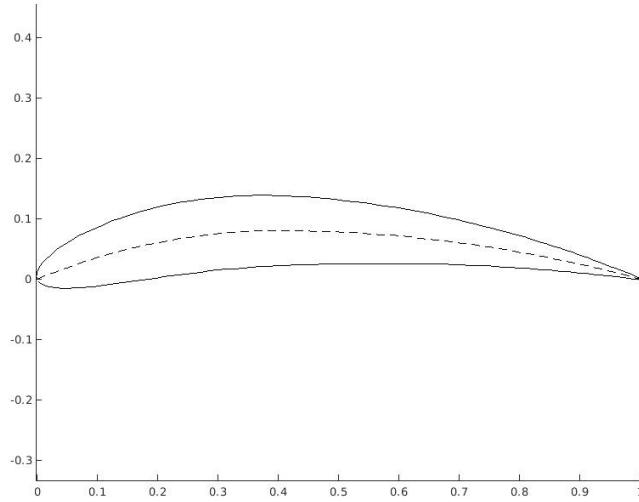


FIGURE (3.1) NACA 8412

3.2 Domain geometry

In order to design a mesh for the numerical resolution, the domain where the equations will be solved must be defined. It must be pointed that all the distances has been expressed as a function of the airfoil chord C .

The NACA 8412 airfoil's leading edge has been placed at the origin of the coordinate axis, and the airfoil has been fixed at zero degree in order to design only one computational domain. As a result, the angle of attack of the case study has been represented by discomposing the freestream velocity in x and y components.

The distance between the leading edge and the inlet following the x axis has been set at $10c$, while the distance between the trailing edge and the outlet has been set at $18c$. In addition, the height of the domain is $16c$, symmetrically distributed in respect to the airfoil. The point A represented in Fig. 3.2, which represents one of the connecting points between the inlet and the outlet, must guarantee that the angle formed by its tangent is larger than the maximum angle of attack studied (15°). For this configuration, the tangent line to the inlet curve at the point A and the horizontal form an angle of 36.87° , complying with the restriction explained. A graphic representation of the domain is presented in Fig. 3.2

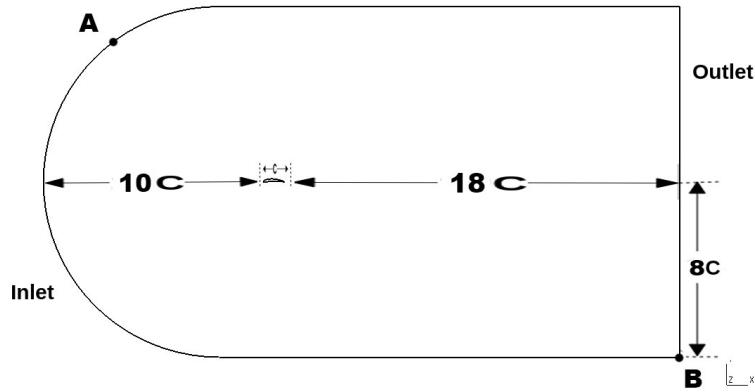


FIGURE (3.2) Domain geometry

3.3 Mesh typologies

In order to justify the decision taken regarding the mesh design, a brief description of different mesh typologies have been presented. There are two main types of meshes: structured or unstructured. Structured meshes are generated by intersecting lines (2 families of lines in 2D), and each mesh point is the result of the intersection of one line of each family. Unstructured meshes are composed of triangles, quadrilaterals or polygons in 2D geometries, and the points are located arbitrarily in the domain. Nevertheless, since both mesh typologies can be implemented using different configurations, a brief definition of the different mesh typologies explained in [5] is presented below:

3.3.1 Structured mesh

The ideal structured mesh would have an equidistant point distribution ($\Delta x = \Delta y$), known as a Cartesian distribution, leading to the highest accuracy possible of the discretised formulas. However, this configuration may considerably increase the computational time due to high mesh accuracy in zones of the domain where it's not necessary (e.g far from the boundary layer). For this reason, non-uniform Cartesian meshes must be considered depending on the geometry studied. This type of mesh allows to progressively modify the mesh spacing, adding accuracy in

zones that require it (e.g. in the boundary layer), but optimizing those regions where less accuracy is required [23]. In addition, non-uniform Cartesian meshes can also be generated by subdividing certain zones of the Cartesian mesh in sub-cells, making it possible to easily refine specific zones of the mesh.

When solid curved surfaces are present, it must be decided whether to introduce a curved mesh to unify it with the solid surfaces (known as "body fitted") or to apply a Cartesian distribution (uniform or non-uniform) with a special treatment of those cells that interfere with the solid surfaces.

There are three main methods for non-body fitted approaches, which are graphically represented in Fig. 3.3:

- **Immersed boundary method:** The Cartesian mesh doesn't suffer any modification on both sides of the surface, but a numerical treatment must be added in the solver to compute the physical boundary conditions.
- **Staircase shape:** The cells in the zones outside the computational domain are removed, and the boundary of the solid surface is approximated with Cartesian cells.
- **Cut-cells:** This method requires a finite volume method, as the cells that intersect with the solid surface are modified with arbitrary shapes adapted to the boundary geometry.

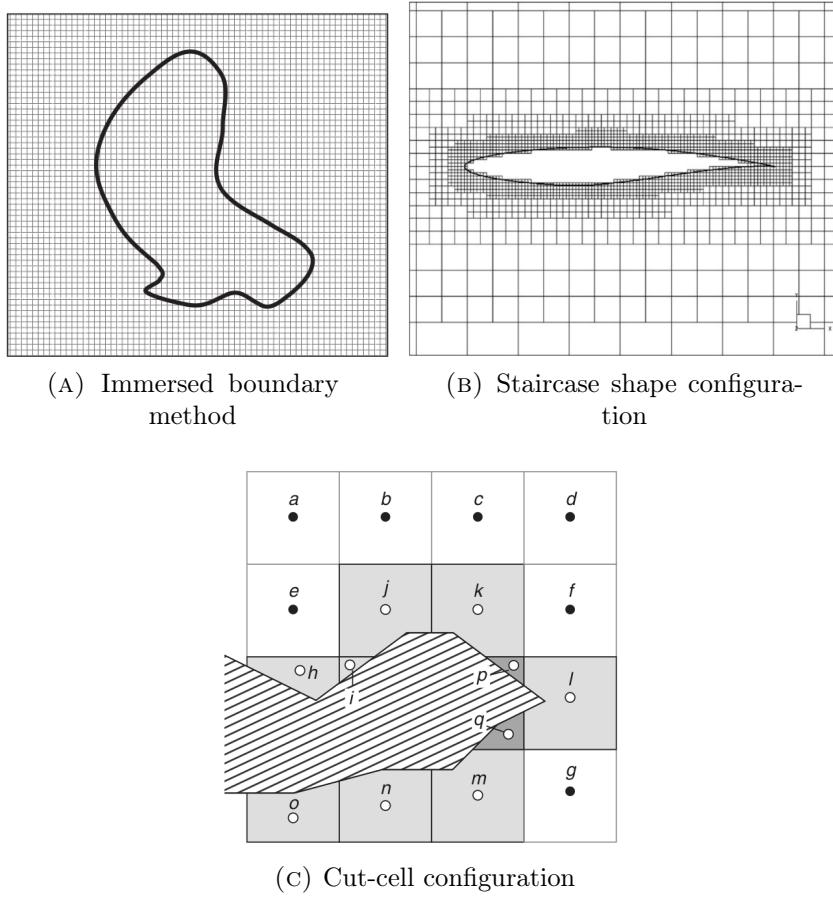


FIGURE (3.3) Non-body fitted mesh approaches [5]

Regarding body-fitted configurations, three main approaches can be followed. A graphic representation of each configuration is presented in Fig. 3.4:

- **H-mesh:** The horizontal and vertical curvilinear lines conforming the mesh follow a pseudo-orthogonal configuration, and lines in both boundaries of the domain are open.
- **C-mesh:** The curvilinear lines surrounding the solid geometry are closed on one side of the domain, but open on the other side. This configuration is especially useful to capture the geometry of a sharp trailing edge, as well as to concentrate lines in the wake zone.
- **O-mesh:** The curvilinear lines completely surround the solid geometry, allowing an accurate distribution of the points located at both leading and trailing edge.

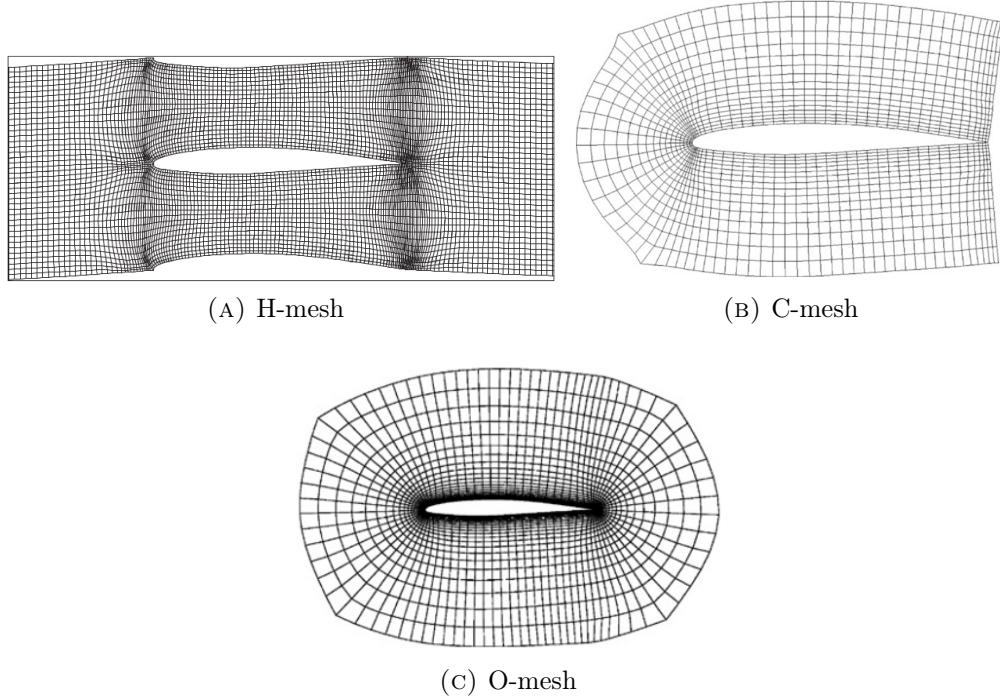


FIGURE (3.4) Body fitted mesh approaches [5]

3.3.2 Unstructured mesh

Despite the fact that with a same point distribution, a structured mesh would have a greater accuracy than an unstructured one, these offer a more flexible mesh adaptation. Unstructured meshes enable to easily refine certain zones independently, as well as solving the body-fitted problems previously discussed, optimizing both accuracy and computational requirements. In addition, the mesh elements can have various geometries, which are listed below:

- Triangle/tetrahedral
- Quadrilaterals and hexahedrals
- Combination of various geometries as tetrahedrals, pyramids and prisms

3.4 Mesh design process

Taking into account the properties described at Sec. 3.3, it has been decided to design an hybrid mesh, including both structured and unstructured configurations. The motivation behind this decision is to remain a balance between accuracy in the inner zones of the domain but also reduce as much as possible the computational resources needed. The process followed to automatically generate the structured and unstructured meshes is explained below.

3.4.1 Structured sub-mesh design

According to the structured mesh properties explained at Sec. 3.3, it has been decided to design two body-fitted structured sub-meshes surrounding the airfoil to guarantee enough accuracy. In addition, and taking into account that the wake region will be later designed using a structured mesh, a C-mesh typology has been chosen. However, taking into account that the accuracy required decreases with the distance in respect to the airfoil, both sub-meshes have a progressive growth: 1% for the one closest to the airfoil and 3% for the one surrounding the first sub-mesh.

In order to simplify the mesh generation and later modifications, a MATLAB script (the same used to discretize the airfoil) present in Appendix ?? has been developed. The script enables to automatically generate the mesh as well as writing it into a .geo file, format required to visualize the mesh in GMSH. A brief description of the process followed is presented below.

To begin with, normal lines to the airfoil are generated with the objective of having an orthogonal mesh in respect with the airfoil. Once they are computed, the points defining the airfoil are re-escalated in order to form the boundaries that will define both structured sub-meshes. It must be pointed that the points generated must follow the normal directions previously obtained to comply with the orthogonality mentioned. The methodology followed is presented in (3.9) and (3.10).

$$x = x_{airfoil} + n_x \cdot t \quad (3.9)$$

$$y = y_{airfoil} + n_y \cdot t \quad (3.10)$$

Where $x_{airfoil}$ and $y_{airfoil}$ represent the points defining the airfoil, n the vector defining the normal direction in respect with the airfoil and t the distance where the final point will be located in percent of the chord. The boundary of the first sub-mesh has been located at 3 cm of the airfoil and the second one at 20 cm. Once the boundaries have been defined, the points are connected using splines. It must be taken into account that, as a C-mesh typology has been chosen, the points of the sub-mesh's boundaries located near the trailing edge needed an special treatment, as this zone should remain open to design a sub-mesh for the wake region. Finally, the points of the airfoil and in the sub-mesh's boundaries have been connected using straight lines, as it's shown in Fig. 3.5.

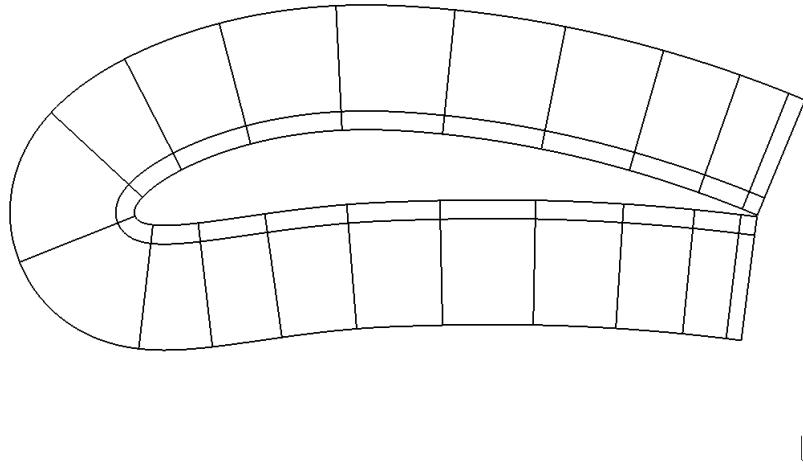


FIGURE (3.5) Connection between boundaries of the sub-meshes

Once the geometry is defined, a study of the first cell height, which will set the number of partitions for the first sub-mesh, must be performed.

First cell height

In order to ensure the reliability of the mesh, the closest zone to the airfoil must be carefully treated. Large changes in velocity and pressure in the normal direction to the surface are observed in this zone. For this reason, the first cell's height becomes one of the main control parameters for our mesh.

Taking into consideration that the aim of the AFC implementation is to modify the behaviour of the boundary layer, the region of the viscous sub-layer ($y^+ < 5$) must be captured (see Chap. 1 Sec. 1.1.1). As a consequence, it has been decided to calculate the first cell height for a $y^+ < 1$. In Chap. 5, different values of y^+ have been studied with the objective of capturing the viscous effects with the less computational cost required. For a fixed y^+ , the first cell height can be obtained with Eq. (1.4), which can be expressed as:

$$\Delta y_1 = \frac{y^+ \nu}{u_\tau} \quad (3.11)$$

In order to evaluate u_τ , presented in Eq. (3.11), the shear stress must be introduced:

$$\tau_w = \frac{1}{2} C_f \rho U^2 \quad (3.12)$$

Where C_f represents the skin friction drag coefficient. A different evaluation C_f must be performed for laminar or turbulent flows, and for external and internal flows. As the present case is focused on a external flow with turbulent behaviour, the coefficient has been computed as:

$$C_f = 0.058 Re^{-0.2} \quad (3.13)$$

Finally, with the height of the first cell, the height of the first sub-mesh and the progressive growth of this region (1% as previously stated), the number of partitions for the inner zone of the structured mesh can be calculated. The resulting mesh for a $y^+ = 1$ is presented in Fig. 3.6.

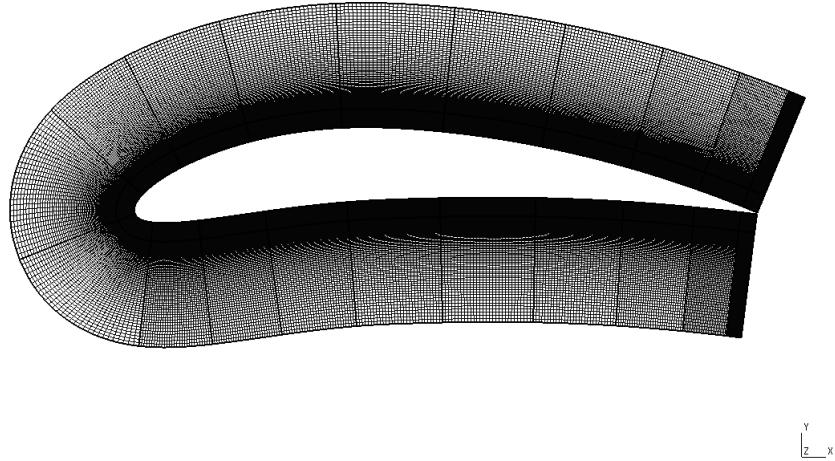


FIGURE (3.6) Mesh design

The mesh present in Fig. 3.6 have been automatically generated by writing its geometrical parameters in a `.geo` file using the MATLAB script present in Appendix ???. The number of partitions of the second sub-mesh must be evaluated in order to ensure a smooth transition from the first to the second sub-mesh. Therefore, the cells in inner cells of the second sub-mesh must be equal or slightly bigger than the outer cells in the first sub-mesh.

3.4.2 Unstructured sub-mesh design

Once the structured sub-meshes surrounding the airfoil have been generated, the unstructured sub-mesh must be designed. The unstructured sub-mesh generation has not been automatized with a script. This is due to the modifications performed to obtain the different meshes studied in Chap. 5 are primarily located in the structured zone.

To begin with, different points have been located in the domain boundaries, and have been connected with arbitrary points on the outer sub-mesh's boundary using straight lines. It must be noted that the wake zone has been designed following this procedure, even though it has been meshed with a structured mesh. The resulting configuration is presented in Fig. 3.7.

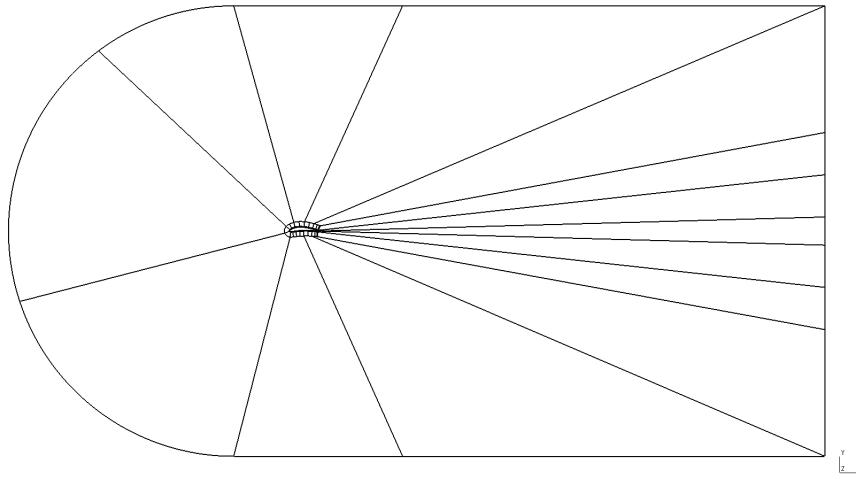
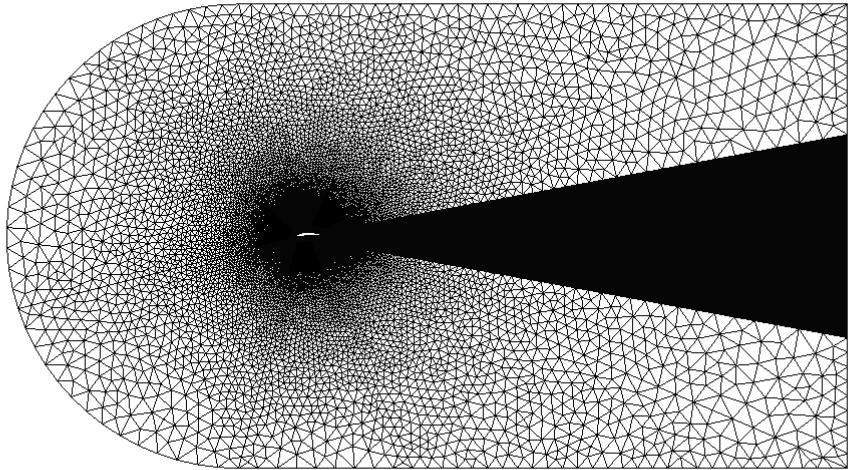


FIGURE (3.7) Unstructured mesh first approach

A smooth change in the cells dimensions in the boundary between the structured and unstructured sub-meshes must be achieved in order to correctly study the flow behaviour. For this reason, an easy modification of the partitions and progressions of the unstructured sub-mesh should be accomplished. Since the modifications in the structured sub-mesh only affect the number of partitions and its progression (not the geometry in Fig. 3.5) , the points generated in the outer boundary of the structured sub-mesh don't change neither their position nor their label. As a result, the geometry represented in Fig. 3.7 remains constant independently from the mesh partition modification, enabling an easy calibration of the sub-meshes transition in the zones where the unstructured and structured sub-meshes are in contact. In addition, the wake zone has been designed with a structured sub-mesh. This is intended to clearly observe the wake behaviour, which can give information about the vortex structures formed and the effectiveness of AFC to modify this vorticular structures. The resultant configurations is presented in Fig. 3.8.



L

FIGURE (3.8) Final mesh design

As it can be seen in the figure above, a smooth transition has been achieved for the zones closest to the airfoil. Considering that less accuracy is required in the outer zones of the domain, the transition between the wake zone and the unstructured sub-mesh around it has not been designed as smooth as the inner zone of the domain. The decrease on the accuracy required can also be observed in the outer zones of the domain, where the triangles forming the unstructured sub-mesh are considerably bigger than those surrounding the airfoil.

3.4.3 Mesh extrusion and export

The next step in the mesh design is to extrude the 2D mesh one layer in the z direction. By doing it, finite volumes are generated, enabling OpenFoam to compute the solution. Even though cells extruded imply a 3D geometry, due to the generation of only one cell in z -direction, the problem can be still treated as 2D. The mesh has been extruded the chord length.

The physical groups present in the domain has been declared. In total, the domain contains seven physical surfaces (*Extrados*, *Intrados*, *TrailingEdge*, *Front*, *Bottom*, *Inlet*, *Outlet*) and 1 physical volume named *Fluid*. This physical groups will later

be used to declare the boundary conditions in OpenFoam, and to visualize the results in ParaView.

Finally, the *.msh* files must be generated based on the *.geo* files obtained following the process presented above. To do so, the resulting mesh has been exported in a *.msh* file directly from GMSH.

In order to check the mesh quality and perform the simulations, it's necessary to transform the *.msh* file into a folder named *polyMesh*. This folder, which has been further explained in Chap. 4 Sec. 4.5, is the OpenFoam folder that contains the geometry of the problem. It's automatically generated inside the OpenFoam *constant* directory using the command *gmshToFoam* in the terminal.

3.4.4 Mesh quality check

With the objective of ensuring correct simulation's results and minimizing the computational time required for the simulations, each mesh has to be checked to assess the mesh's quality. According to [24], the main parameters to guarantee a correct mesh quality are aspect ratio, orthogonality and skewness:

- Aspect Ratio: It's defined as the ratio between the longest and shortest edge's length. For an ideal mesh, its value should be 1. High aspect ratios don't imply that mesh is not correct, but it can drastically affect the convergence time of the results.
- Orthogonality: It's related to the angle formed by the vector connecting two mesh nodes and the normal vector associated with that edge. Its ideal value is 0, and for OpenFoam simulations is recommended not to exceed 70°.
- Skewness: It's defined as the difference between the shape of the cell and the shape of an equilateral cell of equivalent volume [25]. High skewness value can affect considerably the simulation results, since the modeling of the equations to be solved assume that the cells are relatively equilateral/e-quangular.

All this parameters can be visualized using the command *checkMesh*. By doing it, if some of the mesh quality parameters exceed its limit value, a warning is displayed

in the terminal. In addition, if the mesh it's not correctly defined, *checkMesh* fails in the mesh evaluation and the mesh must be modified.

Chapter 4

OpenFoam configuration

The aim of the present chapter is to define OpenFoam and its directories, as well as set the parameters used in the numerical resolution. In addition, the solver used will be briefly introduced. Previously to the development of the chapter, it must be pointed that the folders used to develop [26] has been used as a reference to select certain values or boundary conditions, due to one of the authors is the director of the present Thesis.

4.1 OpenFoam background

According to Chap. 2 Sec. 2.4, OpenFoam is a C++ based toolbox widely used to solve CFD related problems. In order to solve any problem with OpenFoam, it's necessary to previously define three main folders (*0*, *constant*, *system*) which contain the geometry, boundary conditions, solver definition, etc. A brief description of each folder is defined below.

In the *Time folder*, the initial flow and boundary conditions are defined. The most common situation is to name this folder as *0* referring to $t_0 = 0$, although the calculations could began for an initial time value different than 0. At least two files must be present inside the folder:

- **p**: Units, boundary conditions and initial pressure value are defined.

- **U**: Units, boundary conditions and initial velocity value decomposed in its x , y and z components.

Depending on the usage of a turbulence model, additional files must be implemented (see Chap. 1 Sec. 1.1.5).

In the *Constant folder* the geometry and the mesh are defined, as well as the turbulence and physical properties of the flow:

- **polyMesh**: Geometry and mesh definition. Since the mesh has been generated with GMSH, this directory is automatically generated using the *gmshToFoam* command.
- **transportProperties**: Units and freestream value for density and kinematic viscosity. In addition, the transport model for the viscosity must be specified.
- **turbulenceProperties**: This file sets the turbulence model, and it's only required if the solver uses turbulence modelling.

In this *System folder* the parameters related to the solution procedure are defined, and must include at least the following files:

- **controlDict**: Start time, end time, time step and output data parameters.
- **fvSchemes**: Discretisation schemes definition, including gradients, time derivatives, interpolations, etc.
- **fvSolution**: Resolution algorithms for different variables, with its respective tolerance, smoother, and other algorithm control parameters.
- **decomposeParDict**: Decomposes the domain in order to calculate in parallel the different regions (optional).

4.2 Flow solver selection

Depending on the physic and fluidic characteristics of the problem to be studied, a proper flow solver must be selected with the objective of obtaining reliable results.

Due to the high angle of attack used ($\alpha=15^\circ$) a turbulent flow behaviour is expected, which indicates that the turbulence must be contemplated in the solver used. The Mach number of the problem is $Ma = 0.003$, and therefore the case must be treated as incompressible ($Ma < 0.3$). In addition, the transient phase must be captured to properly observe the detachment of the boundary layer.

According to the solvers presented in Chap. 1 Sec. 1.1.4, a selection guided by the flow characteristics has been carried out. The *SimpleFoam* solver has been discarded as it's a steady-state solver, and the transient effects have a great importance on boundary layer detachment and the AFC implementation. In respect with the *PimpleFoam* and *PisoFoam* solvers, the main difference is the time-step's dependence of the results. However, as the studied case is not inherently unstable, the *PimpleFoam* solver has been also dismissed. As a result, *PisoFoam* has been chosen as the solver used.

4.3 Turbulence Model selection

Even though the correct procedure to select a turbulence model should be by numerically validating the best option among different models, it has been considered to be out of the scope. Despite it, a bibliographical approach has been chosen to select the best fitting model.

Taking into account the properties described in Chap. 1 Sec. 1.1.5, a first approach for the turbulence model selection has been carried out. Both *L-VEL* and *algebraic yPlus* have been discarded as they are more suitable for internal flows as well as having a relatively poor accuracy. The four-equation model *v2-f* has been discarded too due to its complexity. Finally, the *low Re k-ε* model has been discarded because of its mesh requirements and the relationship of the convergence with the initial estimation.

In [26], which involves similar case characteristics (high AOA of 14° , $Re=60000$ and AFC purposes) a comparison between Spallart-Allmaras, SST model, LES model and ILES model was carried out. The results showed how the Spallart-Allmaras model was the best-fitting model for the studied case. In addition, the

Director of the Present Theshis is also the director of the reference PhD, and recommended the usage of the Spalart-Allmaras model due to the results obtained in [26]. Moreover, Spalart-Allmaras only adds a variable (modified turbulent kinematic viscosity), which reduces the complexity of the case and offers affordable computational requirements.

4.4 Time directory

As explained in Sec. 4.1, the time directory is named according to the reference start time ($t_0 = 0$ for the present case). As a result, the directory has been named *0*. At least two files must appear on the time directory: *U* file for the initial velocity field and *P* file for initial pressure field. Nonetheless, since Spallart-Allmaras has been used as the turbulence model, the initial field for the turbulent kinematic viscosity (*nut* file) and for the modified turbulent kinematic viscosity (*nuTilda* file) must be contemplated.

4.4.1 Velocity file *U*

The velocity units, initial field value and its boundary conditions must be defined inside the *U* file.

Regarding the units, they are expressed as a vector where every position represents a unit, and its value represents the exponent of the units. In List. (4.1), the second position in the dimensions vector defines the distance [m] and the third position corresponds to time [s].

The airfoil is positioned with an angle of 0° in respect with the horizontal axis inside the domain. Thus, the angle of attack of 15° must be generated by decomposing the freestream velocity on its x and y components (note that the problem is 2D). Taking into account a freestream velocity $U_\infty=1$ m/s and an AoA= 15° , the resulting components of the velocity are presented in List. (4.1).

```
dimensions      [0 1 -1 0 0 0 0];
internalField  uniform (0.9659258 0.258819 0);
```

LISTING (4.1) Velocity field units and internal value

The physical surfaces defining the boundaries in the U file must be equal to those defined in the *polyMesh* folder (see Sec. 4.5). The boundaries are: inlet, outlet, front wall, bottom wall, extrados, intrados and trailing edge. A brief explanation of each velocity boundary condition is explained below for all the physical elements.

Inlet and Outlet

As it can be seen in List. (4.2), the Inlet velocity must be the same as the internal field previously mentioned. The Outlet is defined as *zeroGradient* and, as a consequence, the velocity is constant in the Outlet.

```
Inlet
{
    type          fixedValue;
    value         uniform (0.9659258 0.258819 0);
}
Outlet
{
    type          zeroGradient;
}
```

LISTING (4.2) Inlet and Outlet boundary conditions for velocity

Extrados, Intrados and Trailing edge

In order to comply with the Non-Slip Condition, the velocity on the solid surfaces (in this case the walls of the airfoil) must be set at 0 *m/s*. This condition is expressed in List. (4.3).

```
Intrados
{
    type          fixedValue;
    value         uniform (0 0 0);
}
Extrados
{
    type          fixedValue;
    value         uniform (0 0 0);
}
TrailingEdge
```

```
{
    type          fixedValue;
    value         uniform (0 0 0);
}
```

LISTING (4.3) Extrados, Intrados and Trailing edge boundary conditions for velocity

Front and Bottom

As the problem to be solved has a 2D nature, the Front and Bottom boundaries of the domain must be set as empty. The empty boundary condition represents the directions that are not solved, deactivating the z -component of the velocity. The structure followed is presented in List. (4.4).

```
Front
{
    type          empty;
}
Bottom
{
    type          empty;
}
```

LISTING (4.4) Front and Bottom boundary conditions for velocity

4.4.2 Pressure file P

The same entries as in the U file are written in the pressure field directory. The kinematic pressure units [m^2/s^2] are represented in the dimensions vector in List. (4.5). The internal field has been set to $0\ m^2/s^2$ in order to obtain the resulting pressure field using the relative pressure.

```
dimensions      [0 2 -2 0 0 0 0];
internalField   uniform 0;
```

LISTING (4.5) Pressure field units and internal value

Inlet and Outlet

In List. (4.6) it can be seen how the inlet type has been set to *ZeroGradient*, meaning that the pressure value is constant for this physical entity. The outlet has been set at a fixed value equal to the internal field pressure.

```

Inlet
{
    type      zeroGradient;
}
Outlet
{
    type      fixedValue;
    value    uniform 0;
}

```

LISTING (4.6) Inlet and Outlet boundary conditions for pressure

Intrados, Extrados and Trailing edge

As intrados, extrados and the trailing edge correspond to solid surfaces, the pressure must have a zero gradient in the surface normal direction. This condition can be observed in List. (4.7).

```

Intrados
{
    type      zeroGradient;
}
Extrados
{
    type      zeroGradient;
}
TrailingEdge
{
    type      zeroGradient;
}

```

LISTING (4.7) Intrados, Extrados and Trailing Edge boundary conditions for pressure

Front and Bottom

The same situation as with the velocity (see List. (4.4)) must be applied, setting empty the front and bottom surfaces to solve the problem as 2D.

```

Front
{
    type          empty;
}
Bottom
{
    type          empty;
}

```

LISTING (4.8) Front and Bottom boundary conditions for pressure

4.4.3 Turbulent kinematic viscosity file *nut*

The dimensions for the turbulent kinematic viscosity ν_t are $[m^2/s]$, as it can be seen in the dimensions vector in List. (4.9). For the internal field value, and according to [27], the turbulent kinematic viscosity has been set to $\nu_t = 10^{-20}$.

```

dimensions      [0 2 -1 0 0 0 0];
internalField   uniform 1e-20;

```

LISTING (4.9) Front and Bottom boundary conditions for ν_t

Inlet and Outlet

For both inlet and outlet, the turbulent kinematic viscosity must match with the value given to the internal field. This condition can be observed in List. (4.10).

```

Inlet
{
    type          calculated;
    value         uniform 1e-20;
}
Outlet
{
    type          calculated;
    value         uniform 1e-20;
}

```

LISTING (4.10) Front and Bottom boundary conditions for ν_t

Intrados, Extrados and Trailing edge

The wall condition used for extrados, intrados and trailing edge provides a wall constraint on the *nut* value setting it to 0. In addition, the *nutLowReWallFunction*

showed in List. (4.11) provides an access function to calculate y^+ .

```

Extrados
{
    type          nutLowReWallFunction;
    value         uniform 0;
}
Intrados
{
    type          nutLowReWallFunction;
    value         uniform 0;
}
TrailingEdge
{
    type          nutLowReWallFunction;
    value         uniform 0;
}

```

LISTING (4.11) Extrados, Intrados and TrailingEdge boundary conditions for

ν_t

Front and Bottom

Due to the 2D character of the case, the front and bottom surfaces must be set into empty again, likewise it has been previously done with velocity and pressure.

```

Front
{
    type          empty;
}
Bottom
{
    type          empty;
}

```

LISTING (4.12) Front and Bottom boundary conditions for ν_t

4.4.4 Modified turbulent kinematic viscosity file *nuTilda*

The Spallart-Allmaras model is characterized by solving one additional equation for modified turbulent kinematic viscosity, or $\tilde{\nu}_t$. As a result, the same procedure for velocity, kinematic pressure and turbulent kinematic viscosity has to be followed. The units for the modified turbulent kinematic viscosity are equal to the ones describing ν_t [m^2/s], as showed in List. (4.13). Regarding the internal field, its value is equal to the turbulent kinematic viscosity previously explained.

```
dimensions      [0 2 -1 0 0 0];
internalField  uniform 1e-20;
```

LISTING (4.13) Dimensions and internal field value for $\tilde{\nu}_t$

Inlet and Outlet

The *freestream* condition set for both inlet and outlet provides a mixed condition where the mode of operation can change between fixed (freestream condition) and zero gradient depending on the sign of the parameter [28]. The fixed value set for this condition is equal to the used for the internal field ($\tilde{\nu}_t=10^{20} \text{ m}^2/\text{s}$).

```
Inlet
{
    type          freestream;
    freestreamValue uniform 1e-20;
}
Outlet
{
    type          freestream;
    freestreamValue uniform 1e-20;
}
```

LISTING (4.14) Inlet and Outlet boundary conditions for $\tilde{\nu}_t$

Intrados, Extrados and Trailing edge

Likewise with the turbulent kinematic viscosity for the airfoil's surfaces, $\tilde{\nu}_t$ has to be fixed at 0. However, as the y^+ will be estimated with ν_t , *nutLowReWallFunction* doesn't have to be applied. In this case, it's only necessary to set a fixed value, as it can be seen in List. (4.15).

```
Extrados
{
    type          fixedValue;
    value         uniform 0;
}
Intrados
{
    type          fixedValue;
    value         uniform 0;
}
TrailingEdge
{
    type          fixedValue;
    value         uniform 0;
}
```

LISTING (4.15) Extrados, Intrados and TrailingEdge boundary conditions for $\tilde{\nu}_t$

Front and Bottom

Due to the 2D character of the case, the front and bottom surfaces must be set into empty again, likewise it has been previously done with velocity, pressure and kinematic viscosity.

```

Front
{
    type      empty;
}
Bottom
{
    type      empty;
}

```

LISTING (4.16) Front and Bottom boundary conditions for $\tilde{\nu}_t$

4.5 Constant directory

As explained in Sec. 4.1, the constant folder includes the mesh definition as well as the turbulence and thermophysical settings. Three main entries can be found inside the directory: *Transport Properties* file, *Turbulence Properties* file and *PolyMesh* folder. The two files define the thermophysical properties, and the *PolyMesh* folder contains the files which define the mesh. A brief explanation of this files is presented below.

4.5.1 Transport Properties

In the *Transport Properties* file, the freestream values for both density ρ and kinematic viscosity ν are described, as well as their units. The value selected for both parameters corresponds to the ISA standards at sea level, resulting in $\rho = 1,225 \text{ kg/m}^3$ and $\nu = 1.46 \cdot 10^{-5} \text{ m}^2/\text{s}$, as presented in List. (4.17). A transport model for ν is required. Since air is a Newtonian fluid, the *Newtonian* model

has been used, which considers that ν is constant.

```
transportModel Newtonian;
rho [1 -3 0 0 0 0] 1.225;
nu [0 2 -1 0 0 0] 1.46e-5;
```

LISTING (4.17) Transport Properties

4.5.2 Turbulence Properties

As it's mentioned in Sec. 1.1.5, it has been used a RANS based model. For this reason, the simulation type has been set to RAS (Reynolds Averaged Simulation). In List. (4.18) it can also be seen that the selected turbulence model is the Spalart Allmaras, which has been previously described in Chap. 1 Sec. 1.1.5. Moreover, to compute the turbulence it must be set to on, as well as the option *printCoeffs* in order to observe the model coefficients by the terminal during the simulation.

```
simulationType RAS;
RAS
{
    RASModel SpalartAllmaras;
    turbulence on;
    printCoeffs on;
}
```

LISTING (4.18) Transport Properties

4.5.3 PolyMesh

The *polyMesh* folder describes how the mesh is organized, based on the faces of each cell. Internal faces connect 2 cells, and faces located in the boundaries connect a cell and a boundary patch. In order to set the connectivity within cells, each internal face has assigned an owner cell, and a neighbour cell. Regarding the boundary faces, the connected cell is declared as the owner, and the neighbour is labeled as "-1".

Inside this folder, the following files can be found. The description of each file has been based on [29]:

- **points:** Contains the vectors defining the vertices of the cell. The order of appearance of the vectors describe the numbering given to each vertex.

- **faces**: Defines the faces using the list of the vertices present in the *points* file. The numbering of the faces follows the entries in the file, same as in the point's case.
- **owner**: Contains a list of the owner cell's label. The numbering of the owners is related to the index given to the faces (e.g the first entry in the file is the owner label for face 0).
- **neighbour**: Contains a list of the neighbour cell's label. The numbering is equal to the one used for the owners.
- **boundary**: Contains the patches defining the boundaries. In addition a dictionary is declared using the patch name.
- **Sets folder**: Contains two files related to certain cells definition. One of them is named *Fluid*, as a consequence of defining a physical volume as Fluid. Thus, this file contains those cells that must be treated as fluid. The second file is *highAspectRatioCells*, and contains those cells with large Aspect Ratio.

All this files are automatically generated when using the *gmshToFoam* command. However, the file *boundary* must be modified in order to declare the patch typology of each physical surface. For this reason, only the *boundary* file has been presented.

For each surface, the number of faces (*nFaces*) and the face where it begins (*startFace*), according to the face numbering previously explained, are automatically filled. A brief description of each input is presented below.

Inlet and Outlet

Inlet and Outlet must be declared as patch. This is the common definition of inlet and outlet surfaces, and doesn't contain any geometric information regarding the mesh.

```

Inlet
{
    type           patch;
    physicalType  patch;
    nFaces         66;
    startFace     623600;
}

```

```

    Outlet
{
    type           patch;
    physicalType  patch;
    nFaces        407;
    startFace     623666;
}

```

LISTING (4.19) Boundary Intrados, Extrados and Trailing Edge

Intrados, Extrados and Trailing Edge

Intrados, Extrados and Trailing Edge, presented in List. (4.20), must be declared as wall. This boundary condition defines the physical elements as solid surfaces, and enables to apply the non-slip condition at this surfaces.

```

    Extrados
{
    type           wall;
    physicalType  wall;
    nFaces        270;
    startFace     623060;
}
Intrados
{
    type           wall;
    physicalType  wall;
    nFaces        270;
    startFace     623330;
}
TrailingEdge
{
    type           wall;
    physicalType  wall;
    nFaces        29;
    startFace     624073;
}

```

LISTING (4.20) Boundary Intrados, Extrados and Trailing Edge

Front and Bottom

Front and Bottom of the domain must be declared as empty, likewise they have been declared in files *U*, *P*, *nut* and *nuTilda*. This condition declares the directions that are not solved, deactivating the z component of the flow.

```

Front
{
    type           empty;
    physicalType  empty;

```

```

        nFaces          160557;
        startFace      301946;
    }
    Bottom
    {
        type          empty;
        physicalType empty;
        nFaces          160557;
        startFace      462503;
    }

```

LISTING (4.21) Boundary Front and Bottom

4.6 System directory

This directory contains the files associated with the solution procedure. It must contain at least 3 files: *controlDict*, *fvSchemes* and *fvSolution*. It can contain an extra file named *decomposeParDict*. It must be pointed that this folder has been mainly configured using the OpenFoam set up used in [26], as the director of the cited PhD is the director of the present Thesis.

A brief definition of each file is presented below.

4.6.1 controlDict

In this file, the run control parameters are defined. Observing List. (4.22), it can be seen how the solver must be specified in first place. According to Sec. 4.2, the solver used is pisoFoam.

Regarding temporal parameters, *startTime* has been set at 0 for all the simulations. *endTime* has been assessed using the errors of each simulation in order to guarantee the convergence of the results. However, almost all the simulations have been computed for 30 seconds, as shown in List. (4.22).

One of the main parameters to take into account is *deltaT* due to its influence on the results reliability. In order to select a proper time step, the Courant number (*CFL*) explained in Chap. 2 has been used. *CFL* must not exceed 1 in order to consider the simulation as reliable. As a result, and taking into account that this parameter can be monitored using the terminal, *deltaT* has been evaluated

in the beginning of each simulation in order to guarantee that $CFL < 1$ for all the computational time simulated.

Finally, the other parameters present in List. (4.22) are related to the printing of the temporal parameters.

```

application      pisoFoam;
startFrom        latestTime;
startTime         0;
stopAt           endTime;
endTime           30;
deltaT            0.0002;
writeControl      timeStep;
writeInterval     1000;
purgeWrite        0;
writeFormat       ascii;
writePrecision    8;
writeCompression  on;
timeFormat        general;
timePrecision     8;
runTimeModifiable true;

```

LISTING (4.22) ControlDict file

ControlDict file is also used to include functions that simplify the results extraction. The functions included in the file are summarized below.

- *residuals*: Saves both pressure and velocity residuals for each time value simulated. This data is crucial in order to evaluate the convergence of the results.
- *forces*: Computes the aerodynamic coefficients (C_l and C_d) for each time value simulated. In order to define them, it's necessary to specify the direction of each force, the free-stream density, a reference length (chord) and a reference surface (the product of the chord and the extrusion length). Due to the number of input elements modified, this function has been represented in List. (4.23)
- *fieldAverage1*: Enables to save averaged selected parameters. In this case, velocity and pressure where selected.
- *yPlus1*: Calculates the y^+ in each solid surface. The solution has been used to compare the estimated y^+ with the calculated in order to guarantee that the mesh precision is according to the one needed (see Chap. 5).

- *wallShearStress1*: Computes the wall shear stresses of the problem's surfaces. It must be defined the surfaces where wall shear stresses are computed (Intrados, Extrados and TrailingEdge for the present case).

```

forces
{
    type          forceCoeffs ;
    libs ( "libforces.so" );
    writeControl  timeStep ;
    writeInterval 1 ;
    patches       (" Extrados " " Intrados " " TrailingEdge " );
    rho           rhoInf ;
    log           true ;
    rhoInf        1.225 ;
    liftDir       (-0.258819 0.9659258 0) ;
    dragDir       (0.9659258 0.258819 0) ;
    CofR          (0 0 0) ;
    pitchAxis     (0 0 1) ;
    magUInf       1 ;
    lRef          1 ;
    Aref          1 ;
}

```

LISTING (4.23) Forces function

4.6.2 fvSchemes

In *fvSchemes* file, the numerical algorithms used to solve gradients, derivatives, laplacians, interpolations, etc. are specified. Given the complexity of defining this terms, and as it has been previously mentioned, it has been used the configuration used in [26]. The file configuration is presented in List. (4.24).

```

ddtSchemes
{
    default      Euler ;
}
gradSchemes
{
    default      Gauss linear ;
}
divSchemes
{
    default      none ;
    div(phi,U)    bounded Gauss linearUpwind grad(U) ;
    div(phi,nuTilda) bounded Gauss linearUpwind grad(nuTilda) ;
    div((nuEff*dev2(T(grad(U))))) Gauss linear ;
}
laplacianSchemes
{
    default      Gauss linear corrected ;
}
interpolationSchemes
{
    default      linear ;
}

```

```

}
snGradSchemes
{
    default      corrected;
}
wallDist
{
    method meshWave;
}

```

LISTING (4.24) fvSchemes file

4.6.3 fvSolution

The equation solvers, tolerances and algorithms are defined in *fvSolution* file. Likewise with *fvSchemes*, this file has been based on the one used for [26]. Note that in List. (4.25) it has been specified the solver algorithm used, which is PISO.

```

solvers
{
    p
    {
        solver          GAMG;
        tolerance       1e-06;
        relTol          0.1;
        smoother        GaussSeidel;
    }
    pFinal
    {
        $p;
        tolerance       1e-07;
        relTol          0.1;
    }
    "(U|nuTilda)"
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        nSweeps         2;
        tolerance       1e-09;
        relTol          0.1;
    }
}
PISO
{
    nCorrectors      2;
    nNonOrthogonalCorrectors 1;
    pRefCell         0;
    pRefValue        0;
}

```

LISTING (4.25) fvSolution file

4.6.4 decomposeParDict

This file has been used to decompose the domain in different sub-domains. Each sub-domain is simulated in parallel, reducing the total time required to simulate each case. Note that in List. (4.26) it has been presented only the part of the file where the divisions are represented.

The domain has been divided in 4 parts, each one computed by a core of the AWS cluster rented. In order to specify how the domain is divided, a method must be selected. In this case, it has been used the *simple* method, which decomposes the domain in equal parts according in the directions specified in the corresponding subdictionary *simpleCoeffs* [30]. Thus, observing *simpleCoeffs*, it can be seen how the domain has been divided in 2 parts in the x direction and 2 parts in the y direction.

Finally, the syntax used to run the simulations with the solver previously selected (PISO), but decomposing the domain in the mentioned divisions is: *mpirun -np 4 pisoFoam -parallel*.

```
numberOfSubdomains 4;

method          simple;
simpleCoeffs
{
    n           ( 2 2 1 );
    delta       0.001;
}
```

LISTING (4.26) decomposeParDict file

Chapter 5

Mesh independence test

The aim of this chapter is to study and select the optimal mesh, which ensures that the simulation's results are independent from the mesh density, and minimizes the computational cost. In order to generate the mesh candidates, different y^+ values have been analyzed.

The mesh candidates have been simulated for the case of study properties presented in Chap. 2. The resulting C_l , C_d , aerodynamic efficiency (E), pressure coefficient (C_P) and friction coefficient (C_F) obtained for each mesh have been presented and a comparison has been carried out to select the best mesh.

5.1 Mesh candidates

Following the procedure explained in Chap. 3, section Sec. 3.4.1, four different meshes for y^+ values of 1, 0.7, 0.5 and 0.3 have been generated. The automatic meshing script has been developed with the aim to simplify this process, as manually designing the different meshes would have supposed an extra cost of working time. The four meshes generated are presented in Fig. 5.1

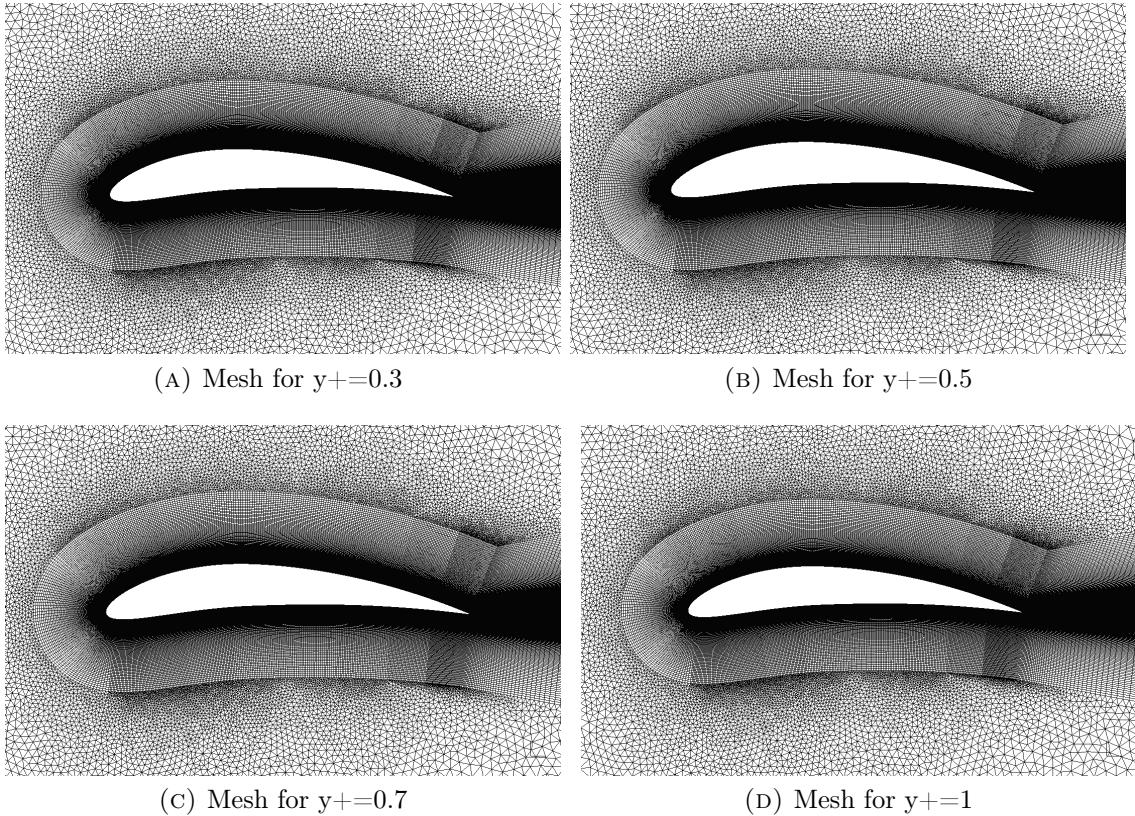


FIGURE (5.1) Studied meshes for the MIT

As it can be seen in Fig. 5.1, the mesh density increases as y^+ decreases. This is due to the relation between y^+ and the first cell height Δy_1 explained in Chap. 3, section Sec. 3.4.1. In Tab. 5.1, the main quality parameters presented in Chap. 3 Sec. 3.4.4 obtained using the *checkMesh* command are presented. Aspect Ratio, skewness and non-orthogonality values presented correspond to the maximum for each mesh.

y^+	N_{cell}	Aspect Ratio	Skewness	Non-orthogonality (°)
1	120879	30.72	0.79	65.13
0.7	139551	30.72	0.79	65.13
0.5	160557	30.72	0.79	65.13
0.3	183897	33.53	0.79	65.13

TABLE (5.1) Mesh quality comparison

As it can be observed, only the number of cells change significantly for the different y^+ values. Aspect ratio, skewness and non-orthogonality have remained constant (except aspect ratio for $y^+=0.3$) for each study case. This may indicate that the lower quality cells are located in the outer zones of the domain or in the wake region. Since cells near the surface suffer changes in their geometry for each y^+ , if the cells with less quality corresponded to those cells, notable changes in aspect ratio, skewness and non-orthogonality would have been observed. The *checkMesh* study hasn't showed any warning or failure, and as a result the four meshes have been considered correct.

5.1.1 MIT criteria followed

In order to select the final mesh configuration, the best procedure would have been to compare the computational results with experimental data, and evaluate the relative error between them. However, no experimental data for the NACA-8412 airfoil at $Re=70000$ were available.

As an alternative procedure, it has been decided to compare the temporal averaged C_l and C_d (and as a consequence, E) obtained for the different candidates. Moreover, C_P and C_F distributions along the chord have been generated to evaluate the results' modification as function of y^+ . Thus, the criteria followed is intended to observe for which y^+ the results don't significantly change, and to evaluate whether the relative error between the results compensates in terms of computational cost.

The four mesh candidates have been simulated for the baseline case conditions using the OpenFoam configuration explained in Chap. 4. So that the results could be considered correct, it's very important to ensure that they have converged. For this reason, the computational time for each simulation have been assessed observing the residuals of pressure and velocity. In a first approach, the meshes were simulated for 20 computational seconds. However, the residuals indicated that the results weren't converged yet, being the computational time raised up to 30 seconds per mesh candidate.

5.2 MIT results

The results after simulating each mesh for 30 computational seconds obtained have been presented in this subsection. The graphics of the temporal evolution of C_l and C_d for $y^+ = 1$, $y^+ = 0.7$, $y^+ = 0.5$ and $y^+ = 0.3$ are presented in Fig. 5.2 and Fig. 5.3 respectively

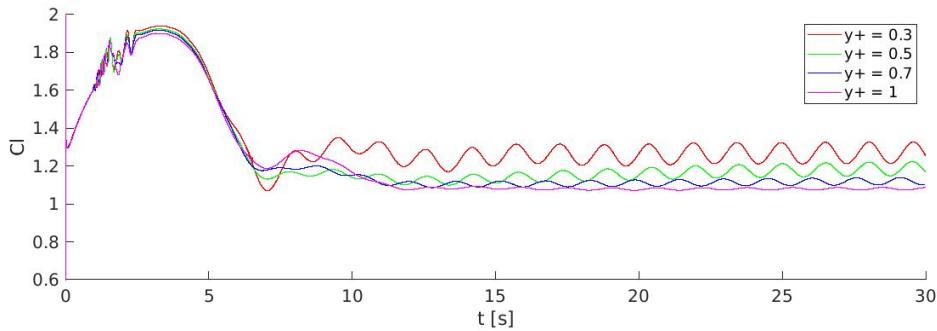


FIGURE (5.2) C_l vs time for each mesh candidate

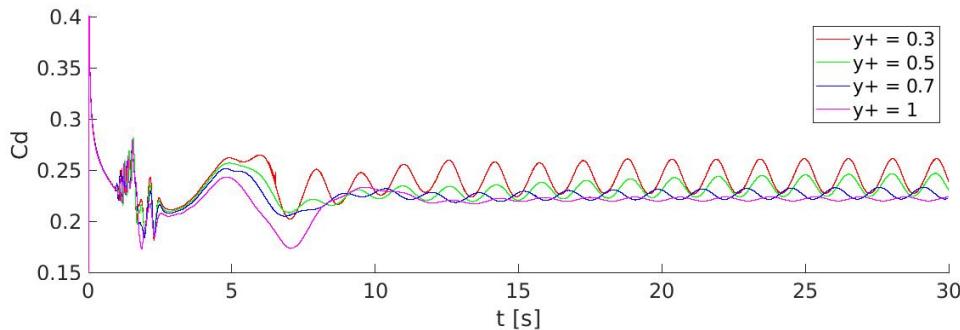


FIGURE (5.3) C_d vs time for each mesh candidate

In the figures presented above, it can be observed how the 10 first seconds are governed by a transient behaviour. After those 10 seconds, and up to the total computational time, the solution for the four cases tend to become more stable, showing a clear oscillation for both C_l and C_d . The amplitude and frequency of the oscillations are related to the vortex shedding produced over the airfoil.

Still related to the oscillatory behaviour of C_l and C_d , it can be observed how its amplitude becomes larger as y^+ is reduced. As an example, for $y^+ = 1$, the time interval from $t = 15s$ to $t = 30s$ shows a nearly constant value of C_l . On the other hand, for $y^+ = 0.3$, and for the same time interval, a clear oscillation of C_l can be observed. This is a consequence of the mesh resolution, as in the most dense mesh

the first cell height is small enough to capture the flow behaviour in the boundary layer, key zone for the lift and drag analysis. However, the less dense meshes cannot capture the flow in the boundary layer, missing its oscillation during time.

5.2.1 Residuals

As it has been previously mentioned, the final residuals for each iteration have been saved in order to observe the convergence of the results. The graphic representation of U_x , U_y and P residuals for $y^+ = 1$, $y^+ = 0.7$, $y^+ = 0.5$ and $y^+ = 0.3$ are presented in Fig. 5.4, Fig. 5.5, Fig. 5.6 and Fig. 5.7 respectively.

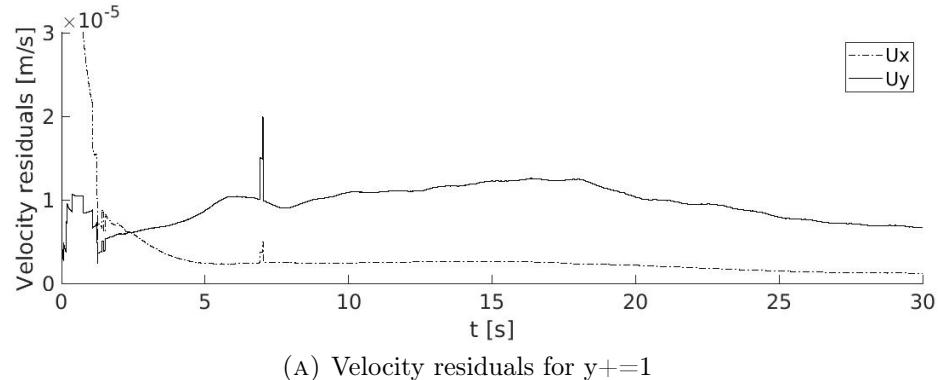
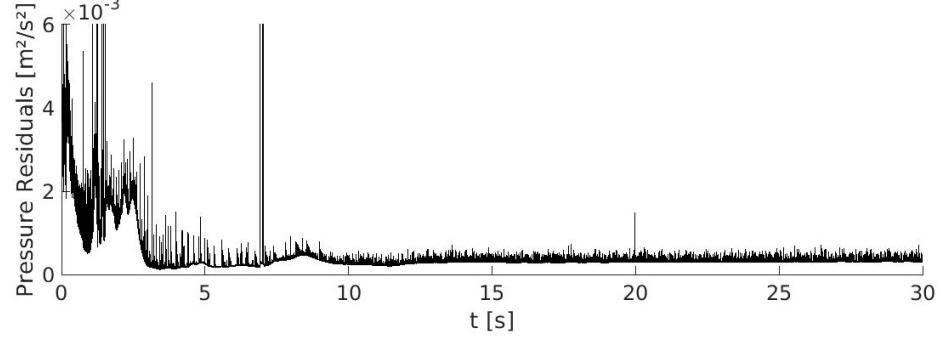
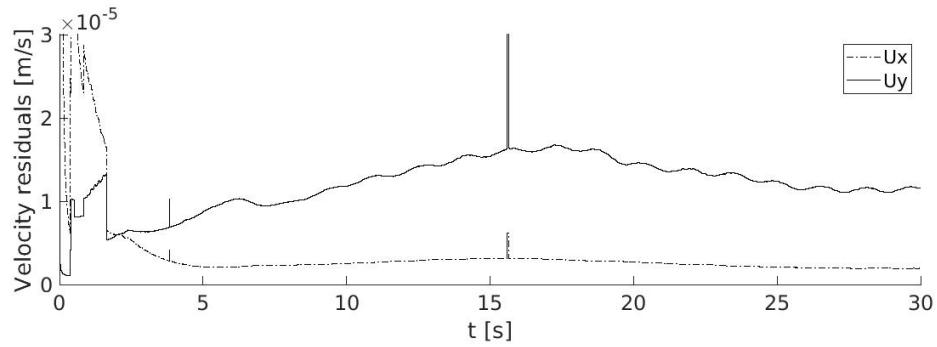
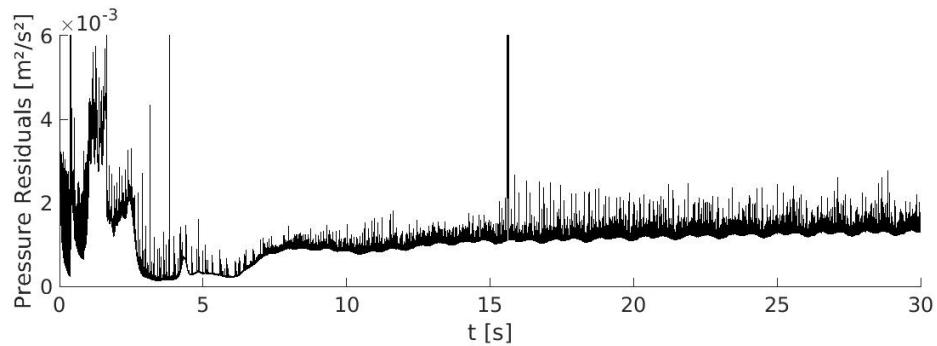
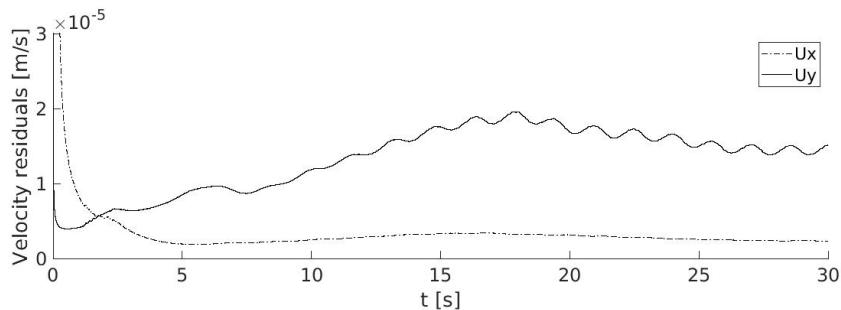
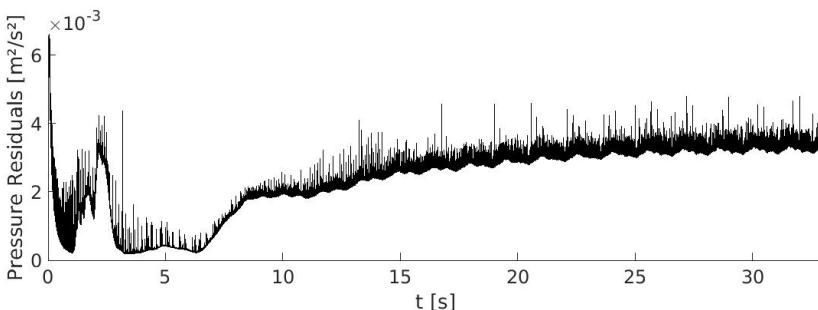
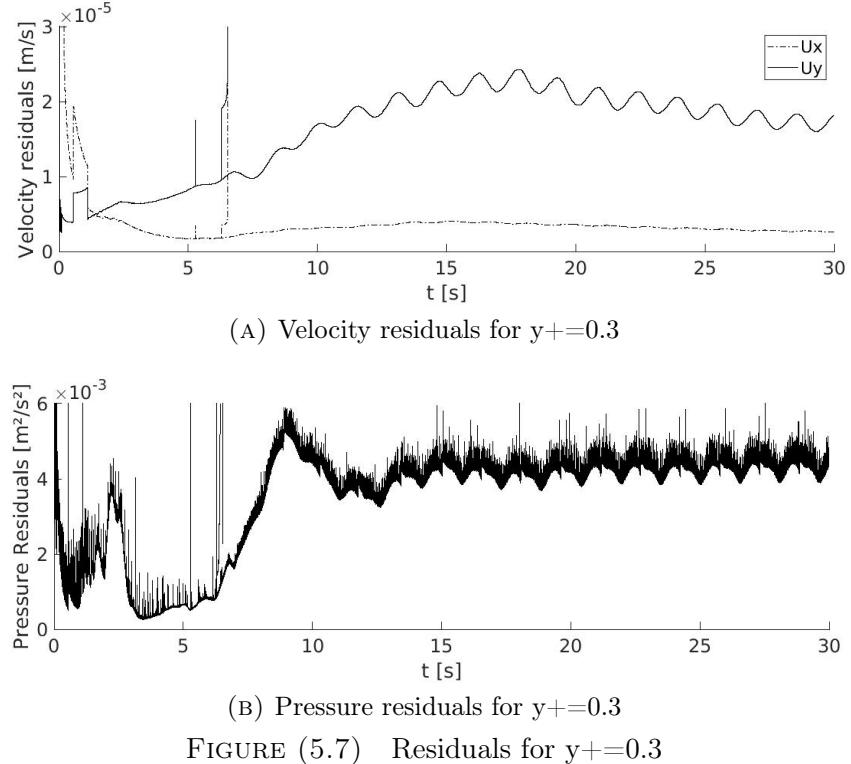
(A) Velocity residuals for $y^+ = 1$ (B) Pressure residuals for $y^+ = 1$

FIGURE (5.4) Studied meshes for the MIT

(A) Velocity residuals for $y+=0.7$ (B) Pressure residuals for $y+=0.7$ FIGURE (5.5) Residuals for $y+=0.7$ (A) Velocity residuals for $y+=0.5$ (B) Pressure residuals for $y+=0.5$ FIGURE (5.6) Residuals for $y+=0.5$

FIGURE (5.7) Residuals for $y^+=0.3$

Even though some residuals still show a non-constant value for $t=30s$ (e.g in Fig. 5.4), it has been accepted the solution as converged due to the small variation of the residuals for the mentioned time value. The reason why the results have been accepted is due to computational and time restrictions. Given the time available for a bachelor degree thesis, the convergence of the results had to be simplified to comply with the terms given by the university. In addition, C_l and C_d graphs previously presented showed a constant oscillation for both coefficients, which can confirm that the solution is nearly converged.

In order to consider the results reliable, it's recommended to have the velocity residuals in the order of magnitude of 10^{-5} , or below. This conditions has been accomplished as the graphs presented confirm. Moreover, for all the y^+ values studied, the U_y residuals are approximately an order of magnitude larger than U_x . The reason of this difference is that a larger modification of the vertical velocity in respect to the horizontal velocity is observed in the boundary layer along time.

5.2.2 Efficiency comparison

The results presented above have shown how both C_l and C_d suffer a variation along time, due to its value oscillates with a given amplitude and frequency. Thus, time averaged values of the aerodynamic coefficients as well as the E have to be assessed to select the optimal mesh selection. Time averaged coefficients have been calculated for the last 10 seconds of each simulation, with the objective of computing only the converged solution, and discard the transient values. Moreover, 10 different C_l and C_d values for different time values have been used to compute their time averaged value. In order to select the sample data, C_l graphs have been used as reference to use significant time values reflecting the variation of the aerodynamic coefficients with time. The resulting time averaged values for C_l , C_d and E for each y^+ mesh are presented in Tab. 5.2. The number of cells of each mesh has also been included in Tab. 5.2 with the objective of simplifying the comparison between the different candidates. In addition, the calculated y^+ for the extrados has been included to evaluate the accuracy of each mesh.

y^+	N_{cell}	C_l	C_d	E	Calculated y^+
1	120879	1.0797	0.2218	4.8679	0.9127
0.7	139551	1.1153	0.2267	4.9197	0.6544
0.5	160557	1.2267	0.2393	5.1262	0.4728
0.3	183897	1.2692	0.2441	5.199e	0.2887

TABLE (5.2) Mesh Independence Test

As Tab. 5.2 indicates, the aerodynamic coefficients tend to stabilize as y^+ decreases, which was anticipated in Sec. 5.1.1. This tendency can be confirmed with the relative error between candidates. For instance, the relative error between $y^+=0.3$ and $y^+=1$ is 14.93% for C_l and 9.14% for C_d . On the other hand, the relative error between $y^+=0.3$ and $y^+=0.5$ is 3.35% for C_l and 1.97% for C_d . The y^+ calculated has been assessed for the upper side of the airfoil due to its importance in the later AFC implementation. Regarding the calculated y^+ values for extrados, it can be observed how the values obtained are slightly lower than the estimated value used to calculate the first cell height. This results confirm that

the first cell's height for the most dense meshes are correct in order to capture the viscous sub-layer.

The minimum relative error calculated indicates how there's still some mesh dependency on the results, and less dense meshes should be studied to observe if this mesh dependency is canceled for the $y^+=0.3$ candidate. However, as the relative error between the aerodynamic coefficients for $y^+=0.3$ and $y^+=0.5$ is relatively small, and the computation time for $y^+=0.3$ (approximately seven days) is nearly the double of the computational time for $y^+=0.5$ (approximately four days with the same conditions), the results have been accepted as representative enough for the detail required in a bachelor degree thesis.

Other parameters that can indicate which mesh candidate can be accepted for the later studies are the C_P and C_F distributions. These coefficients have a great impact on the result analysis of the present Thesis, as they will have a great importance in the AFC implementation and the analysis of the aerodynamic efficiency improvement. For this reason, the pressure coefficient along the chord is presented in Fig. 5.8, as well as the friction coefficient along the chord in Fig. 5.9.

Regarding the pressure coefficient comparison in Fig. 5.8, it can be observed how the results in intrados are very similar for each mesh candidate, despite that $y^+=1$ and $y^+=0.7$ show a slightly lower C_P value for approximately the 80% of the chord. A more notable difference can be observed in extrados, where C_P progressively grows as y^+ decreases. In addition, Fig. 5.8 shows how the C_P in extrados for $y^+=0.3$ and $y^+=0.5$ extrados are very similar along the whole chord, indicating that a certain degree of independence of the solution has been achieved for $y^+=0.3$ and $y^+=0.5$.

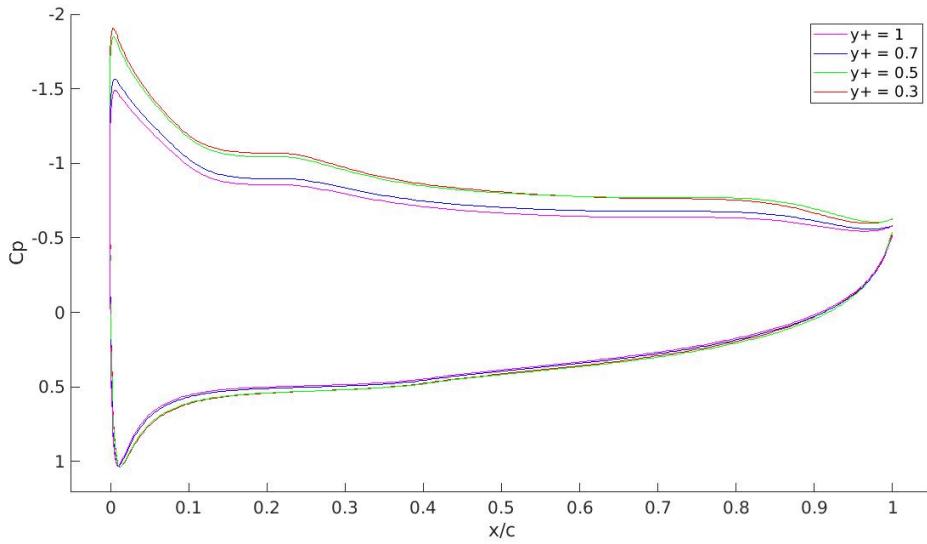


FIGURE (5.8) Pressure coefficient comparison

Due to the later usage of C_F , this coefficient has been studied only in the upper surface of the airfoil. In Fig. 5.9 it can be observed how C_F doesn't significantly change for the different meshes studied. A more extensive analysis of the friction coefficient graph has been carried out in Chap. 6, since it's a key parameter to design the synthetic jet position.

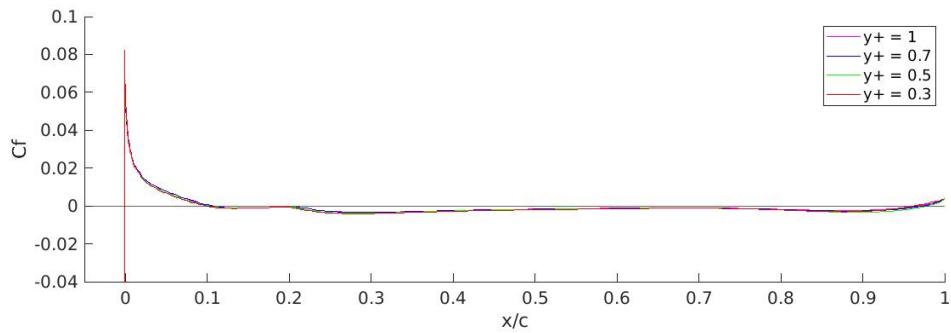


FIGURE (5.9) Friction coefficient comparison

5.3 Baseline case selection

Observing Tab. 5.2, the main candidates to be selected as the baseline case correspond to $y^+ = 0.3$ and $y^+ = 0.5$. For both cases, the calculated y^+ is very close to the estimated y^+ , ensuring the condition of $y^+ < 1$. Even though C_l and C_d present

greater values for $y^+=0.3$, the difference between $y^+=0.5$ and $y^+=0.3$ are $\Delta Cl=-0.0425$ and $\Delta Cd=4.8e^{-3}$ (considering $y^+=0.3$ as the reference). In addition, the difference of cells between the two main candidates is 23340 cells. Even though it could seem as a small difference, the time expended to simulate $y^+=0.3$ (7 days approx.) supposed nearly the double of time than for the $y^+=0.5$ case (4 days approx.).

Regarding Fig. 5.8, the difference between $y^+=0.3$ and $y^+=0.5$ only can be noticed between the LE and $x/C=0.3$. However, this difference can be neglected due to the high similarity of the results. In Fig. 5.9, no difference can be noticed between the C_F distribution for $y^+=0.3$ and $y^+=0.5$.

All things considered, the mesh with $y^+=0.5$ has been selected as the used one. Moreover, the results obtained has been used as the baseline case for later comparisons. The main factor that has influenced the selection is the computational time required for $y^+=0.3$ to simulate. Even though it shows better results, the project has a maximum length of four months, requiring some simplifications in order to comply with the schedule given. In addition, C_l and C_d differences between the $y^+=0.3$ and $y^+=0.5$ meshes are relatively small, indicating that the solutions for $y^+=0.5$ can be used to extract conclusions about the flow behaviour.

The velocity field for $y^+=0.5$ has been presented in Fig. 5.10 with the objective of analyzing the flow behaviour in the baseline case. It can be observed how the separation point is located very near to the LE. After the separation point, the boundary layer is detached and a vortex covering almost all extrados appears in the downstream direction. Thus, the later AFC implementation is intended to modify the vortex shedding generated, decreasing its size.

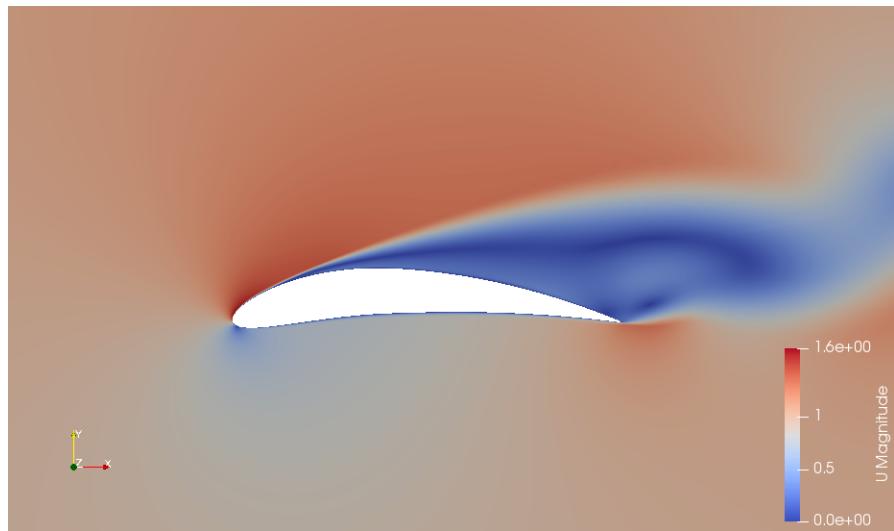


FIGURE (5.10) Velocity field for the basline case

Chapter 6

AFC implementation

The aim of this chapter is to design the synthetic jet using the baseline case results previously obtained. The design of the jet begins by studying its location over the airfoil. To do so, it's necessary to identify the separation point of the boundary layer, which has been obtained using the friction coefficient graphic representation presented in Chap. 5. Once the jet position has been determined, the mesh must be modified to simulate its impact on aerodynamic efficiency.

The next step is to design the AFC velocity signal. As it has been stated in Chap. 1, a ZNMF approach will be followed, and as a consequence a periodic signal of velocity in respect to time must be applied. Thus, the amplitude of the signal as well as its frequency have been assessed.

Finally, the OpenFoam files have been modified to introduce the periodic input and to configure the jet typology as a physical surface.

6.1 Jet location

One of the key parameters when implementing AFC techniques is the position of the jet. Since AFC is intended to influence on the generation of vorticular structures, the slot where the jet is placed must be located upstream in reference to the separation point. The objective of this placement is to prevent the generation

of a huge vortex over the airfoil, breaking it by the periodic injection and suction of energy when it's generated.

In Chap. 5, a first approach of the separation point has been done using the baseline case's velocity field. However, a much precise technique is required to locate the separation point, as the jet position is highly dependant on it. For this reason, the separation point must be assessed using the C_F obtained for the baseline case, which is presented in Fig. 6.1. It must be noted that the graphic represents the time averaged values of C_F , following the same procedure used in Chap. 5 to generate both C_P and C_F comparison graphics.

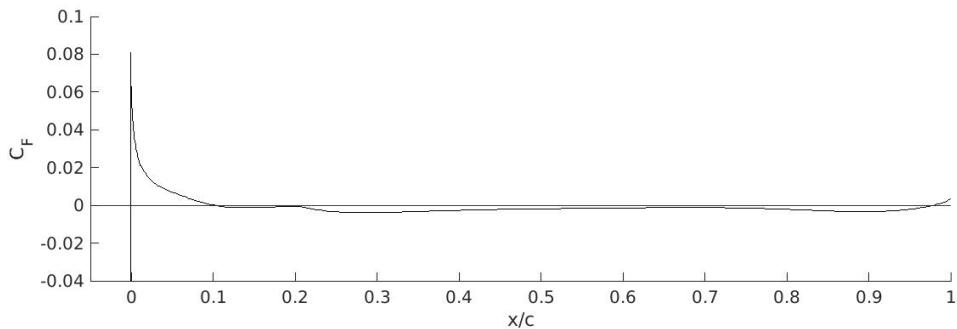


FIGURE (6.1) C_F of the baseline case

As it can be seen in Fig. 6.1, the separation point is located at $x/C=0.1045$ (approximately at the 10% of the chord), considerably close to the leading edge. This is a consequence of the high AoA used (15°), and of the geometry of the airfoil selected, as it shows a pronounced curvature near the LE. At the separation point, C_F begins to adopt negative values, and as a consequence the shear stresses acting over the surface are negative too. This means that, due to the adverse pressure gradient, the flow direction has been inverted leading to the flow separation.

Taking as a reference [26], the synthetic jet has been placed at a location of $0.01C$ upstream of the flow separation point. Moreover, the slot width has been set at $0.01C$, from approximately $x/C=0.08$ to $x/C=0.09$. In order to respect the original geometry, two points of the airfoil already generated have been used to numerically define the slot, more exactly $P1=(0.0839,0.0786)$ and $P2=(0.0935,0.0831)$. The resulting geometry is presented in Fig. 6.2.

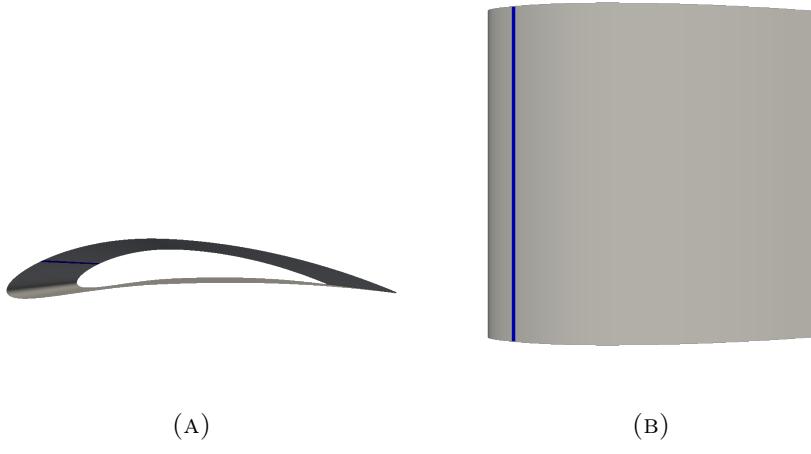


FIGURE (6.2) Jet positioning over the airfoil

6.1.1 Mesh modification

Taking into account the slot definition presented above, it is necessary to modify the geometry and mesh with the objective of simulating the AFC cases, and to add accuracy in the region near the synthetic jet.

As it's been previously mentioned, two points defining the airfoil (P1 and P2) have been used as slot delimiters. An analogue procedure as presented in Sec. 3.4.1 in Chap. 3 has been followed to generate the normal lines for the slot mesh. Note that a relation of orthogonality was created within the points defining the airfoil and the points defining the boundaries of the structured sub-meshes. As a result, only by connecting the correct points, the normal lines defining the jet's mesh has been generated (see Chap. 3). In addition, and to declare the jet as an independent physical surface, a single spline must define the slot length. The resulting geometry to be meshed is presented in Fig. 6.3

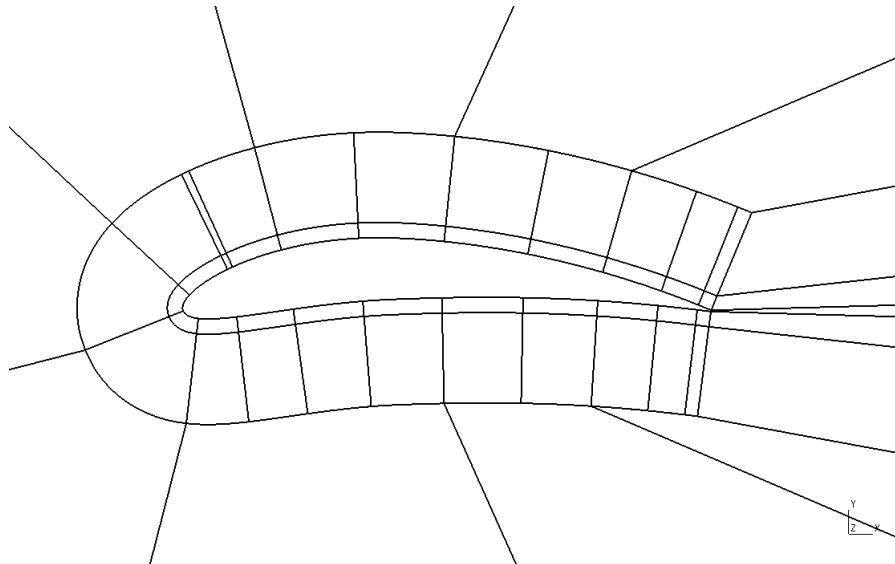


FIGURE (6.3) Mesh for AFC implementation

A more fine meshing is required in the slot region due to the quick variation of the flow properties happening at this zone. Because of the neighbours of the slot region must be structured too, both the number of partitions and the growth rate have been kept constant for the normal lines (76 partitions with a 1% growth rate for the inner structured sub-mesh and 80 partitions with a 3% growth rate for the outer structured sub-mesh). Thus, the partitions of the splines defining the slot has been increased in respect with the neighbour regions, resulting in 15 partitions with no progression rate. The modified mesh is presented in Fig. 6.4

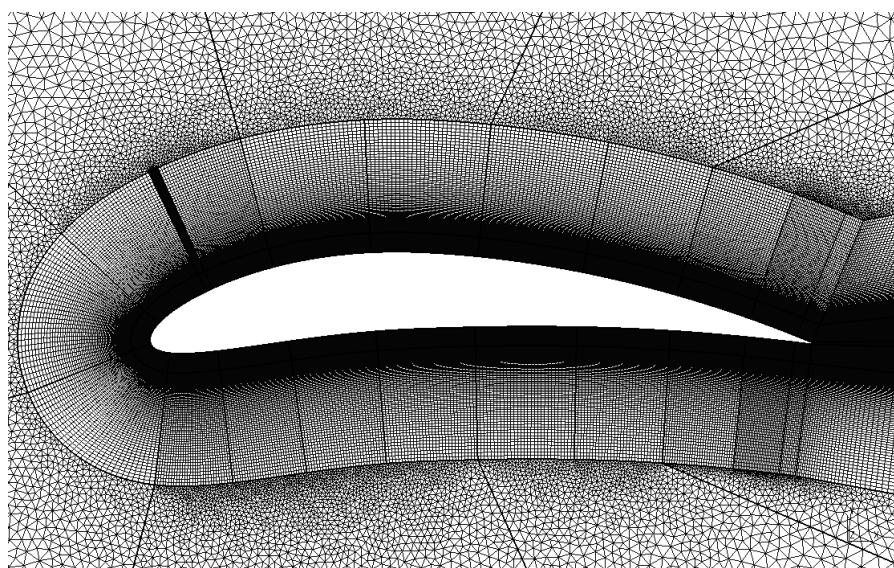


FIGURE (6.4) Scheme of the mesh for AFC implementation

6.2 AFC velocity input

ZNMF synthetic jets are characterized by injecting the same quantity of fluid in the blowing and suction phases (see Chap. 1). For this reason, a sinusoidal velocity input must be designed with the objective of injecting and extracting moment from the boundary layer. As a result, the signal will have the form of Eq. (6.1).

$$U_{jet}(t) = A \sin(2\pi f t) \quad (6.1)$$

Even though the amplitude and frequency will be modified to optimize the jet impact on Chap. 7, an initial approach must be carried out to define the procedure followed as well as estimating a reference value for A and f .

6.2.1 Forcing frequency

The introduction of periodic forcing, and consequently introducing a certain frequency to the input signal, is aimed to modify the vortex shedding by having smaller vortex in extrados. In order to do so, the vortex shedding frequency must be assessed by analyzing its frequency spectrum. Following the methodology presented in [31], a Fast Fourier Transform for the lift coefficient of the baseline case has been performed. The resulting spectrum representation is presented in Fig. 6.5

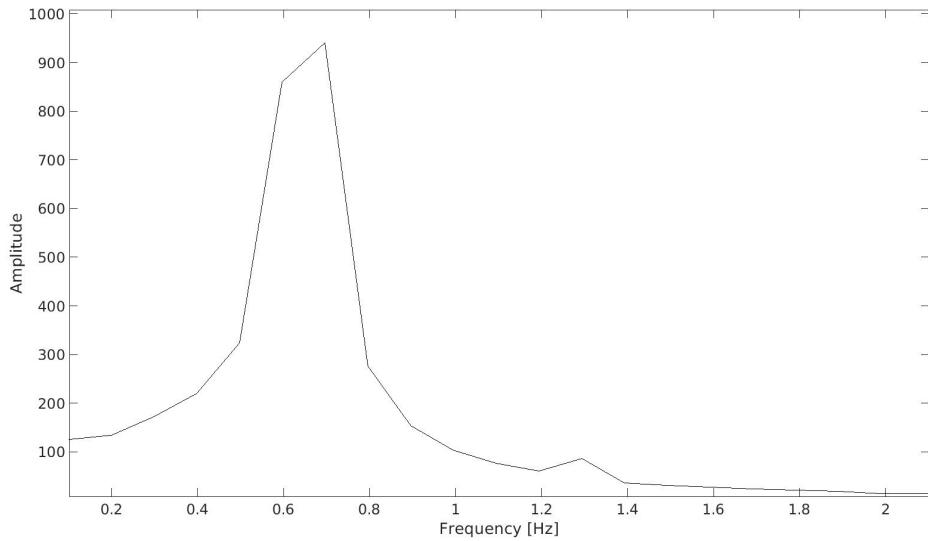


FIGURE (6.5) FFT of lift coefficient

A noisy spectral signal was expected, taking into consideration that the present case is based on turbulent flows. However, as the solution for C_l has not fully converged, the vortex shedding frequency cannot be completely determined. If the solution had converged, the natural frequency would appear as a thin peak. Therefore, it can be stated that the results are not statistically fully reliable, and the computational time should be increased as previously commented in Chap. 5. Moreover, and regarding the noisy signal expected, the lack of more peaks could indicate that the other frequencies detected have a significant lower value than the vortex shedding frequency.

All things considered, the vortex shedding frequency has been considered to be $f_0 = 0.6965$ Hz. This estimation will be validated in Chap. 7, where the results for the given frequency will be evaluated. In addition, a set of non dimensional frequencies F_+ (see Chap. 2, Sec. 2.3) will be assessed. These studies will show if the natural frequency chosen fits the behaviour expected.

6.2.2 Velocity amplitude

The amplitude of the velocity signal is a function of the maximum velocity of the flow when ejected/suctioned from the jet. In order to determine the maximum jet

velocity, the momentum coefficient presented in Chap. 2, Sec. 2.3 must be studied. Thus, a different U_{max} is obtained for each jet moment coefficient studied.

When the maximum velocity of the jet is obtained, the amplitude can be calculated by obtaining the mean value of a semi-period (see Fig. 6.6). The velocity presented in the figure corresponds to $C_\mu = 0.01$, a jet inclination angle of $\theta=30^\circ$ and the jet width previously explained. Note that, as the jet is from the ZNMF class, the mean value for the complete period would be 0, as the suctioning and blowing phases have the same temporal length.

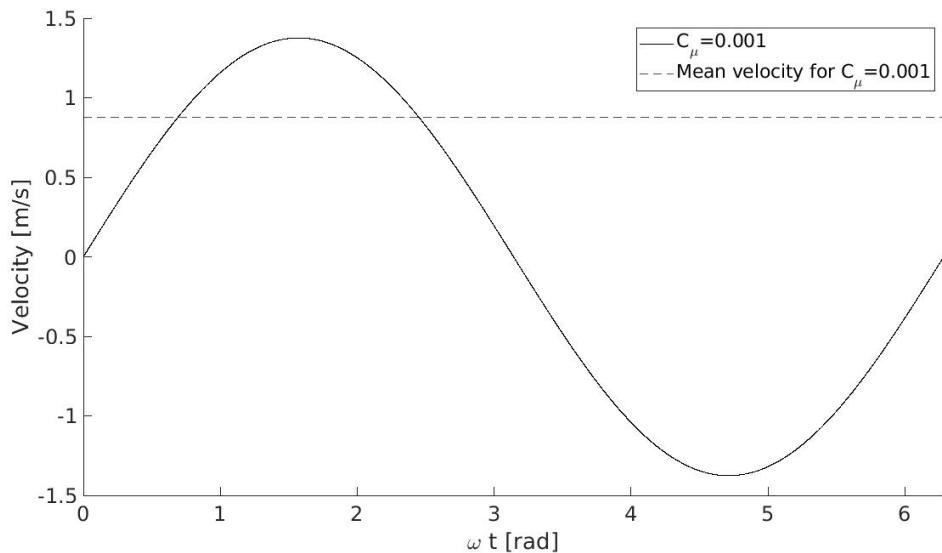


FIGURE (6.6) Umean graphic representation

In Chap. 7, different moment coefficients and jet inclination will be evaluated. As a result, the mean velocity will be modified for each studied case.

6.3 OpenFoam set up for AFC

In order to introduce both the excitation frequency and the signal amplitude in OpenFoam to simulate different AFC cases, both *constant* and θ folders must be modified.

In first place, the slot has been declared as a physical surface named *Jet* in the *polyMesh* folder inside the *constant* directory. According to the surface definition presented in Chap. 4, the Jet surface must be declared as *patch*.

Once it has been declared, its treatment is analogue to the one followed for the inlet for all the files present in the θ folder, excepting the U file. The U file for $C_\mu = 0.01$, with $F^+ = 1$, and $\theta = 30^\circ$. is presented in List. (6.1).

```

dimensions      [0 1 -1 0 0 0 0];
internalField   uniform (0.9659258 0.258819 0);
boundaryField
{
    Inlet
    {
        type          fixedValue;
        value         uniform (0.9659258 0.258819 0);
    }
    Outlet
    {
        type          zeroGradient;
    }
    Intrados
    {
        type          fixedValue;
        value         uniform (0 0 0);
    }
    Extrados
    {
        type          fixedValue;
        value         uniform (0 0 0);
    }
    TrailingEdge
    {
        type          fixedValue;
        value         uniform (0 0 0);
    }
    Front
    {

```

```

        type           empty;
    }
Bottom
{
    type           empty;
}
Jet
{
    type           uniformFixedValue;
    uniformValue   sine;
    uniformValueCoeffs
    {
        frequency     0.6965;
        amplitude     1.0591;
        scale         (0.9947  0.1026  0);
        level         (0 0 0);
    }
}

```

LISTING (6.1) U file modification for AFC implementation

As it can be seen in List. (6.1), the velocity of the *Jet* boundary must be defined as a sine, with its value fixed for all the surface. In order to define the sinusoidal behaviour, four parameters must be declared: frequency, amplitude, scale and level. The frequency introduced must be the dimensional value (in Hz), and amplitude corresponds to the mean velocity previously introduced. Scale is related to the jet inclination, and must be set according the x and y components (note that the problem is 2D) of a unit vector with the inclination angle referenced to the surface tangent. A graphic representation of the vector defining the scale value for $\theta = 30^\circ$ is presented in Fig. 6.7. Finally, level is used to introduce an offset to the input signal and, as a result, modifying the suction and blowing phases. As the present Thesis is focused on ZNMF actuators ,the blowing and suction must be equal to

the function semi period. Consequently, no offset must be introduced.

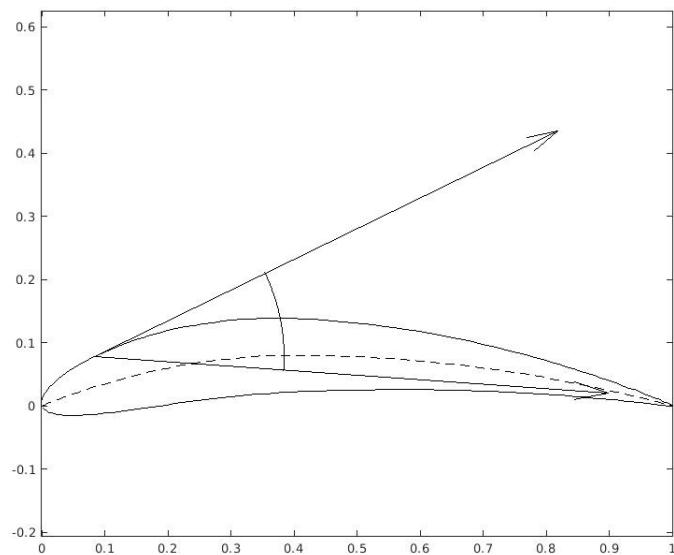


FIGURE (6.7) Angle between vectors defining scale

Chapter 7

AFC Results

The aim of this chapter is to obtain the optimal AFC configuration to maximize the aerodynamic efficiency of the airfoil.

The results extraction has began with a study of the moment coefficient and jet inclination influence on both lift coefficient and drag coefficient. Three jet inclination angles θ have been studied. For each θ , five moment coefficient C_μ have been introduced and evaluated to obtain the optimal θ and C_μ configuration among those studied.

Six non-dimensional forcing frequencies $F+$ have been introduced to the optimum θ and C_μ configuration. The aerodynamic efficiency variation for different forcing frequencies have been assessed and the maximum aerodynamic efficiency has been obtained.

A discussion of the results has been carried out, taking into account the time averaged values for C_l , C_d and E , the convergence of the results and both the pressure coefficient and friction coefficient distributions along the chord.

7.1 AFC results

In order to evaluate the influence of AFC on aerodynamic efficiency, three design variables have been studied. Those variables, used in previous studies as [32], are the jet inclination angle θ , the jet moment coefficient C_μ and the non-dimensional forcing frequency $F+$. It must be taken into account that the jet position and the slot width would also affect the efficiency of the actuator. However, a further study of these parameters have been considered out of scope.

To simplify the results evaluation, different values of θ and C_μ have been studied in first place to obtain a first approach of the optimal configuration. Moreover, it has been considered that this variables will have a bigger impact on E than $F+$, which will be evaluated when the final results are obtained. Once the previous configuration has been simulated, different $F+$ have been introduced to the optimal θ and C_μ case to observe the influence of the forcing frequencies and obtain the final jet configuration.

Each case has been simulated with OpenFoam for 30 computational seconds using the AWS cluster, as it has been previously done for the baseline case. However, a evaluation of the errors must be carried out to ensure that the computational time chosen converges the solution. The baseline case results were $C_l=1.2267$, $C_d=0.2393$ and $E=5.1267$. Thus, this values are expected to be improved with the AFC implementation.

7.1.1 θ and C_μ analysis

According to Chap. 2, the jet inclination angle θ is referenced to the tangent to the surface. Taking this into account, three θ values of 20° , 30° and 40° have been simulated. The inclination angles have been selected according to the advise of the Thesis' director. In addition, for each θ value, five C_μ of 0.001, 0.003, 0.005, 0.007 and 0.01 have been studied. The range of C_μ values have been selected based on previous validated studies [33] [34].

A comparison between the results obtained for the different simulation is presented in Tab. 7.1. It must be pointed that both C_l and C_d presented in the table below

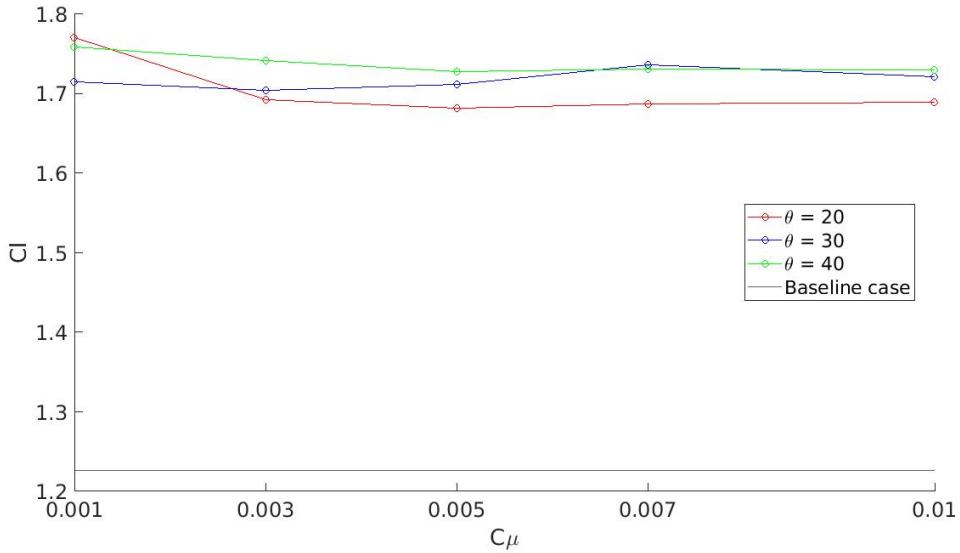
correspond to the coefficients' time averaged values, following the same procedure explained in Chap. 5.

θ	C_μ	C_l	C_d	E
20°	0.001	1.7704	0.1824	9.7061
	0.003	1.6925	0.1586	10.6715
	0.005	1.6823	0.1517	11.0896
	0.007	1.6877	0.1552	10.8744
	0.01	1.6893	0.1570	10.7599
30°	0.001	1.7147	0.1485	11.5496
	0.003	1.7041	0.1478	11.5326
	0.005	1.7122	0.1451	11.7987
	0.007	1.7363	0.1634	10.6249
	0.01	1.7210	0.1631	10.5537
40°	0.001	1.7587	0.1621	10.8512
	0.003	1.7421	0.1413	12.331
	0.005	1.7273	0.1471	11.7460
	0.007	1.7309	0.1439	12.0285
	0.01	1.7304	0.1531	11.3007

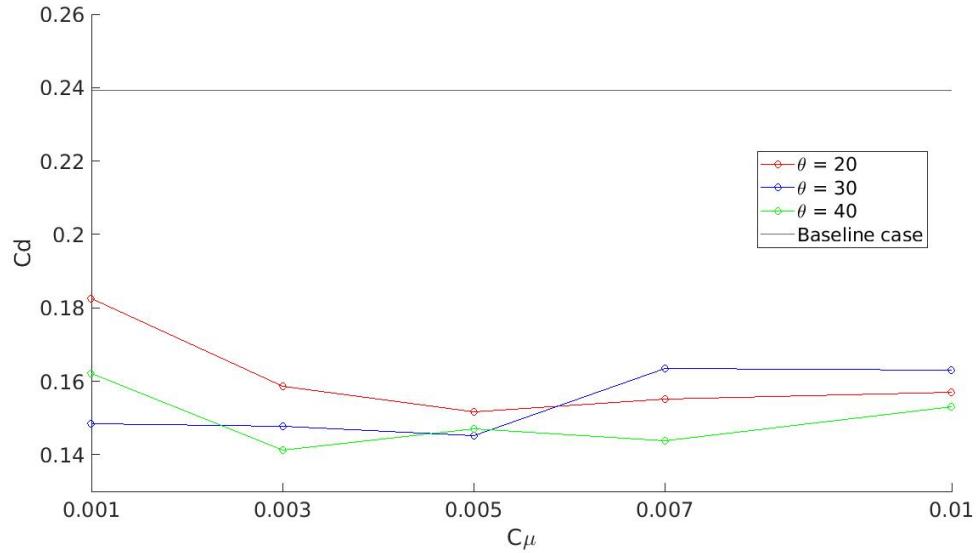
TABLE (7.1) AFC results comparison

As it can be seen in Tab. 7.1, the maximum E is achieved for $\theta=40^\circ$ and $C_\mu=0.003$. In order to simplify the results analysis, the behaviour of C_l , C_d and E as a function of C_μ for each θ studied have been presented in Fig. 7.1, Fig. 7.2 and Fig. 7.3.

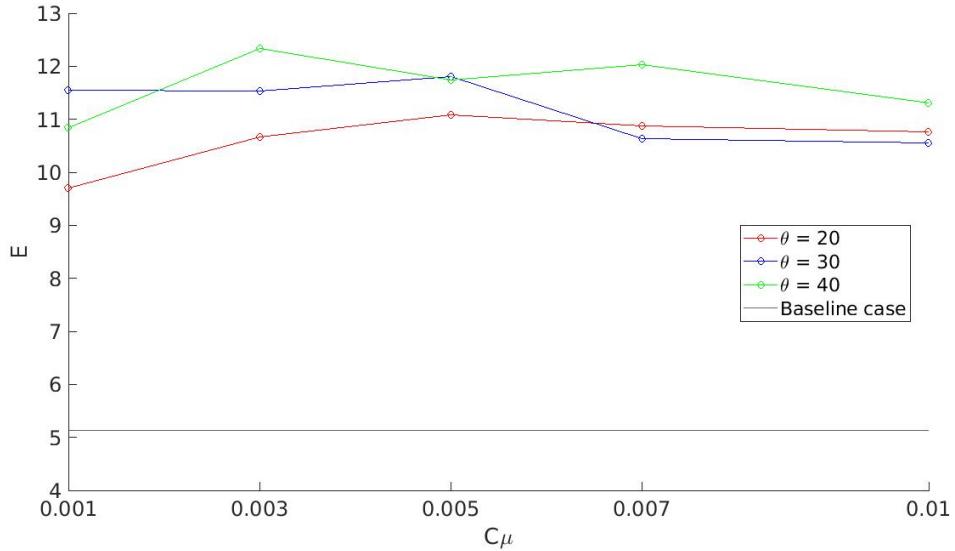
Observing Fig. 7.1, it can be stated that the C_l obtained for all the θ studied have considerably increased in respect with the baseline case. This variation on C_l verifies the procedure followed, as one of the objectives of the present Thesis is to increase the lift coefficient. Even though $\theta=20^\circ$ shows the highest C_l for $C_\mu=0.001$, it decreases for the next C_μ , presenting the lowest C_l values. Regarding $\theta=30^\circ$, the C_l tends to increase with C_μ , reaching its peak at $C_\mu=0.007$, and then it slightly decreases at $C_\mu=0.01$. Finally, $\theta=40^\circ$ present a progressive decrease for $C_\mu=0.001$, $C_\mu=0.003$ and $C_\mu=0.005$. After this tendency, C_l tends to stabilize at $C_l=1.7$ approximately.

FIGURE (7.1) C_l comparison for the different θ and C_μ studied

Regarding Fig. 7.2, the C_d obtained for all the θ values studied are considerably lower than the C_d for the baseline case. Thus, the procedure followed is now again validated, as the reduction of drag is one of the aims of the Thesis. A similar behaviour than the one previously observed for C_l can be observed in the C_d comparison. Again, for $\theta=20^\circ$, C_d tend to decrease for the first 2 C_μ values, and then it stabilizes. Related to $\theta=30^\circ$, a similar behaviour as with C_l is observed too. For the lowest C_μ values, C_d is stable at $C_d=0.15$ approximately, and then increases for $C_\mu=0.007$ and $C_\mu=0.01$ up to 0.16 approximately. Finally, the main differences in the behaviour between C_l and C_d corresponds to $\theta=40^\circ$. For C_l , it was the more stable case, but C_d presents fluctuations around $C_d=0.15$ approximately. Its lowest value is achieved at $C_\mu=0.003$, and corresponds to $C_d=0.1413$.

FIGURE (7.2) C_d comparison for the different θ and C_μ studied

Regarding Fig. 7.3, E for all θ and C_μ has increased considerably. This is a direct result of the previous two figures, as C_l increased and C_d decreased in respect with the baseline case. The maximum aerodynamic efficiency has been obtained for $\theta=40$ and $C_\mu=0.003$, with $E=12.331$. Comparing this case with the baseline case, it supposes a $\Delta C_l=0.51154$, $\Delta C_d=-0.098$, and $\Delta E=7.2043$. Consequently, the results have been considered satisfactory in order to proceed with the F+ study.

FIGURE (7.3) E comparison for the different θ and C_μ studied

The comparisons of C_l , C_d and E show that a wider ranger of C_μ should be studied in order to fully observe its influence on these coefficients. The behaviour for the three coefficients to consider the analysis complete should be a progressive growth for the first C_μ values presented, peaking in a certain C_μ which would indicate the optimum configuration for each θ angle. After the peak, the coefficients should decrease indicating that the optimum configuration has already been achieved.

7.1.2 F+ analysis

Even though the variation of θ and C_μ have shown an improvement of the 140.53% of the aerodynamic efficiency, the modification of F+ is intended to identify the optimum F+ which maximizes C_l and minimizes C_d . The F+ values studied are 0.5, 1, 2, 3, 4, 5. The chosen F+ range have been selected according previous validated studied [26]. In addition, even though it's no included in the cited paper, F+=0.5 has been studied to observe how the vortex shedding (and as a consequence C_l and C_d) is affected for an F+ lower than the unit.

As it has been previously mentioned, the case where different F+ have been applied corresponds to the optimum case obtained in Sec. 7.1.1 ($\theta=40$ and $C_\mu=0.003$). After simulating each F+ case for a computational time of 30s, the time averaged C_l and C_d values in Tab. 7.2 have been obtained

C_μ	F+	C_l	C_d	E
0.003	0.5	1.6473	0.2063	7.9853
	1	1.7421	0.1413	12.3331
	2	1.8075	0.1289	14.0214
	3	1.8272	0.1120	16.3160
	4	1.8147	0.0942	19.2649
	5	1.8220	0.1054	17.2829

TABLE (7.2) F+ comparison

As shown in Tab. 7.2, the optimum F+ case corresponds to F+=4. For this non-dimensional forcing frequency, $C_l=1.8147$, $C_d=0.0942$ and E=19.2649. As a result, an improvement of E in respect with the optimum C_μ and θ of the 56% have been

achieved. In addition, in comparison with the same case ($F+=1$), $\Delta C_l=0.0726$ and $\Delta C_d=-0.0471$.

With the objective of simplifying the results evaluation, the evolution of C_l , C_d and E as function of $F+$ have been represented in Fig. 7.4, Fig. 7.5 and Fig. 7.6.

Observing Fig. 7.4, it can be stated that $F+=0.5$ showed the lowest C_l of those obtained for the rest of $F+$. For this $F+$, C_l is lower than those obtained in Sec. 7.1.1. C_l tends to maintain relatively constant around $C_l=1.8$ for $F+$ values greater than 1. The maximum C_l , corresponding to $C_l=1.8272$, has been obtained for $F+=3$.

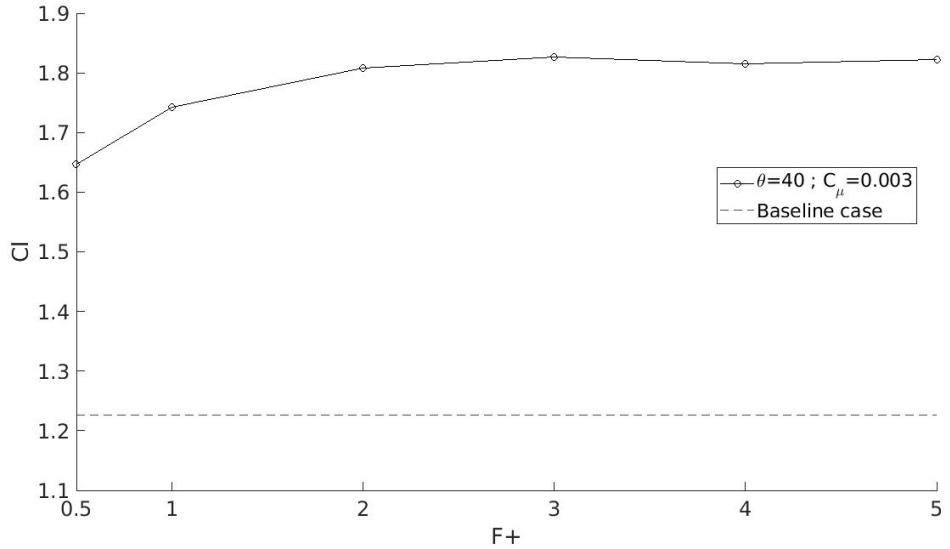
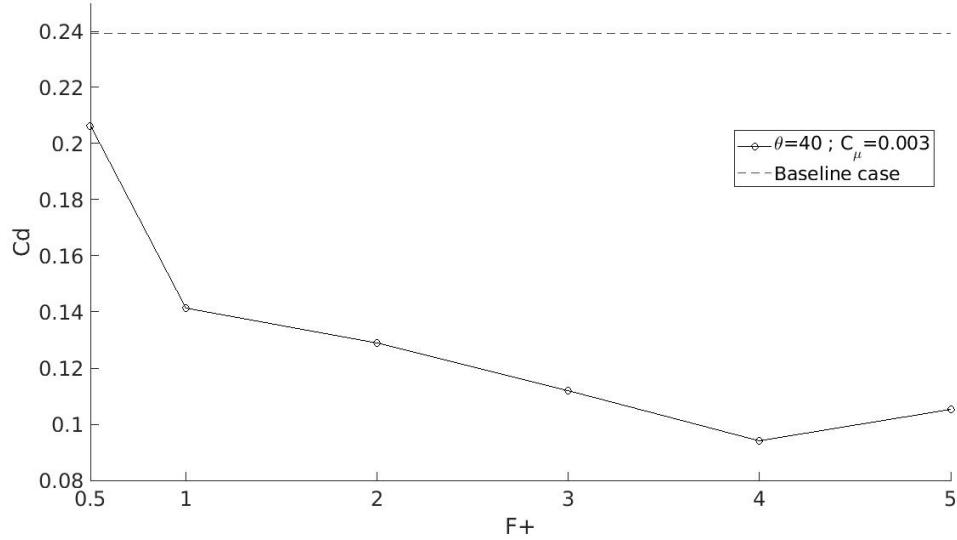
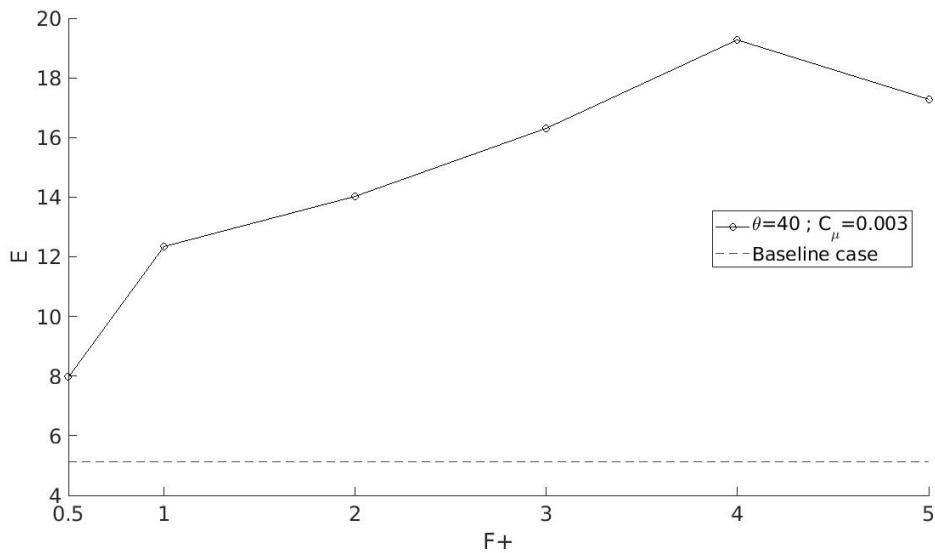


FIGURE (7.4) C_l comparison for the $F+$ studied

Observing Fig. 7.5, it can be seen how the variation of $F+$ have had a greater impact on C_d than on C_l . Similar to Fig. 7.4, $F+=0.5$ presents the poorest improvement, with $C_d=0.2063$. For $F+=2, 3$ and 4 , C_d has a decreasing trend, reaching a minimum of $C_d=0.0942$ for $F+=4$. However, for $F+=5$, an increasing trend of C_d is observed.

FIGURE (7.5) C_d comparison for the $F+$ studied

Regarding Fig. 7.6, it can be observed how the optimum $F+$ case is achieved for $F+=4$. This is a direct consequence of the decrease on C_d , taking into account that Fig. 7.4 showed a much stable behaviour of C_l , remaining almost constant for $F+$ values greater than 1. As a result, it can be stated that the modification of the vortex structures caused by $F+$ [35] have had a positive impact on drag reduction.

FIGURE (7.6) E comparison for the $F+$ studied

All things considered, a further analysis of the results obtained for the optimum case is presented in Sec. [7.2](#)

7.2 Optimal configuration analysis

The maximum E have been achieved for $\theta=40^\circ$, $C_\mu=0.003$ and $F+=4$. However, in order to guarantee that the results are reliable, the simulation has been extended up to 50 computational seconds. This increase on the computational time is intended to observe the residuals evolution, and confirm whether the solution has converged or a longer simulation is required. Observing Fig. [7.7](#) and Fig. [7.8](#), a repeating pattern in both pressure and velocity residuals can be observed from $t=25s$ to $t=50s$. Even though the simulation is not completely converged, given the computational cost required to increase the computational time and the time available to develop a bachelor degree thesis, the results have been accepted as correct.

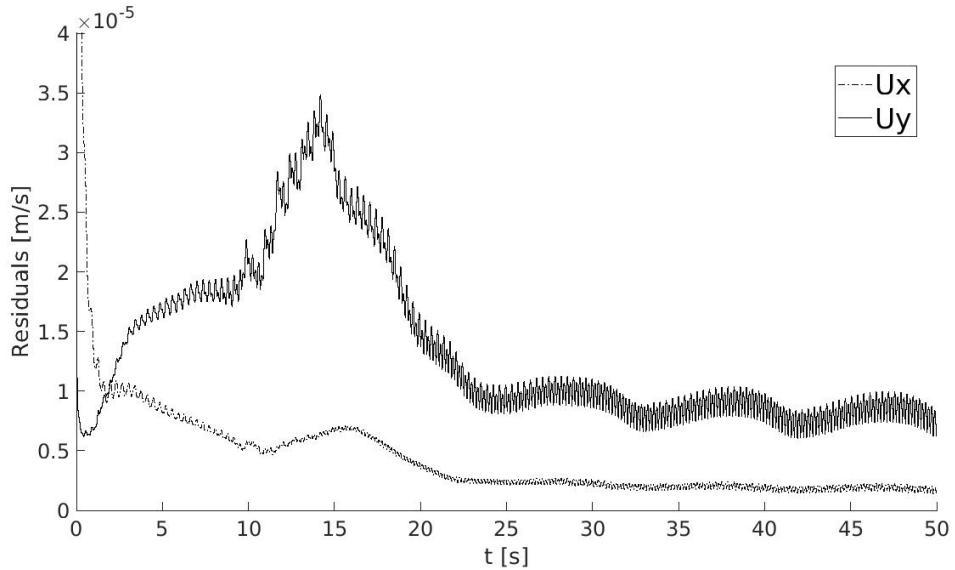


FIGURE (7.7) Velocity Residuals for $F+=4$, $\theta=40^\circ$ and $C_\mu = 0.003$

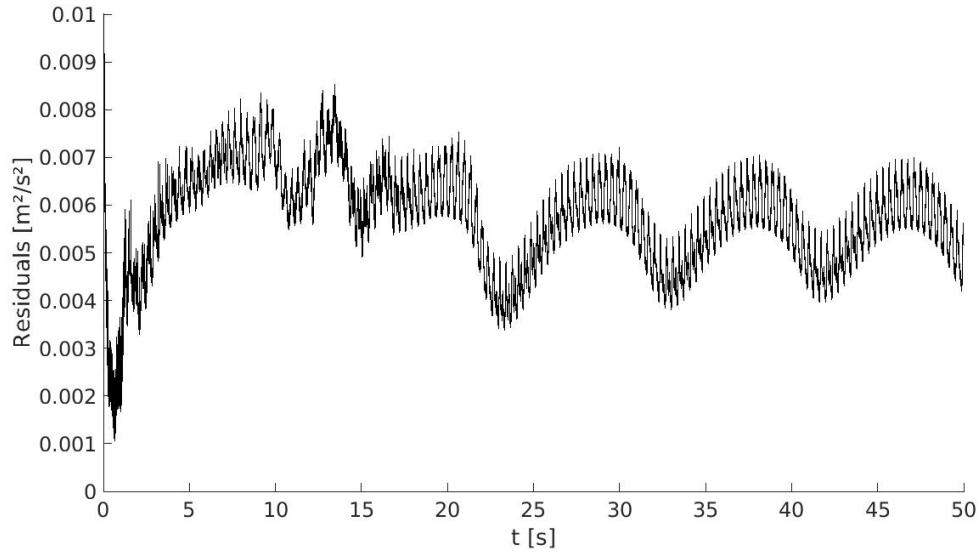
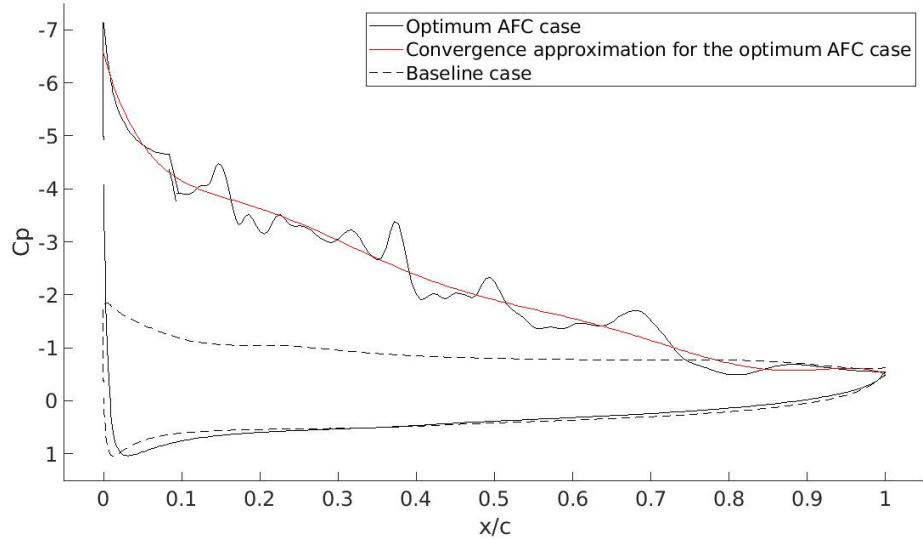


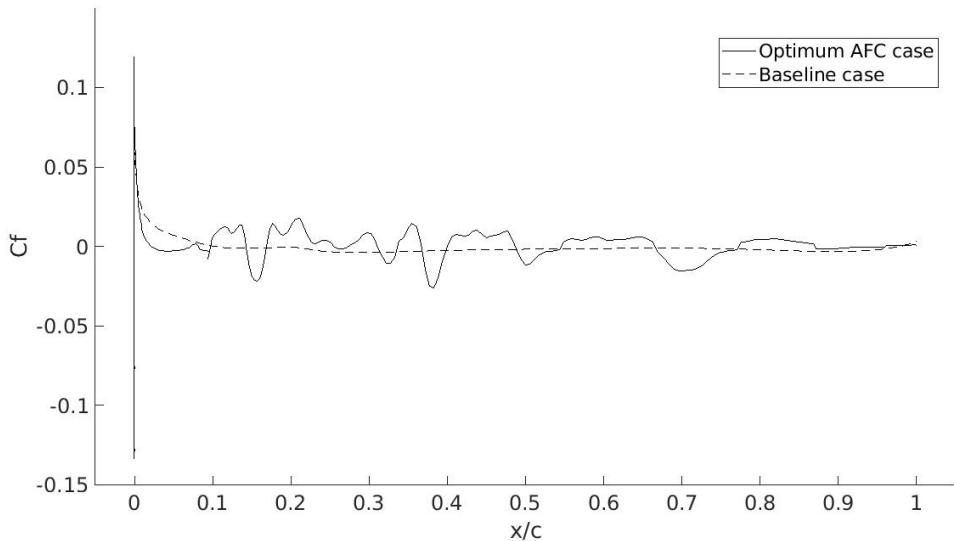
FIGURE (7.8) Pressure Residuals for $F+=4$, $\theta=40^\circ$ and $C_\mu = 0.003$

Fig. 7.9 shows a considerable increment of C_P on extrados compared with the baseline case. This behaviour was expected, as lift has been increased a 48% in respect with the baseline case. In addition, the high irregularity denotes that small vortex have been generated in extrados, and move downstream. The suction peak has remained very close to the LE. All things considered, it can be stated that the C_P distribution is in accordance with the results expected, and with the theoretical hypothesis.

Even though the C_P graph has been obtained using time averaged C_P values, its shape is highly irregular. This indicates that the solution is still statistically not fully reliable, and the simulation should be extended in order to find a fully converged solution. However, as it has been previously explained, the pressure and velocity errors for the last 25 seconds show a nearly constant pattern. Thus, the solution has been accepted and the behaviour of the vortex shedding can be already evaluated. In order to extract conclusions, an approximation of C_P on extrados has been plotted to represent the converged solution.

FIGURE (7.9) C_P for $F+=4$, $\theta=40^\circ$ and $C_\mu = 0.003$

The extrados C_F comparison between the optimum AFC case and the baseline case is presented in Fig. 7.10. Once again, the irregularity on the results show the appearance of small vortex on extrados, which moves in the downstream direction. Fig. 7.10 it's a very good example to observe the influence of AFC on the vortex shedding, as in the baseline case only one vortex was generated for $x/C=0.1$ approximately, and finished near the TE. In conclusion, AFC implementation has broken this huge vortex, generating smaller ones relaminazing the boundary layer, and improving the momentum transmission among layers.

FIGURE (7.10) C_F for $F+=4$, $\theta=40^\circ$ and $C_\mu = 0.003$

In Fig. 7.11 and Fig. 7.12, the temporal evolution of C_l and C_d for the optimum case and the baseline case have been presented. In Fig. 7.11 it can be observed how the amplitude for the optimum configuration has been reduced in respect with the baseline case. For the optimum configuration, the amplitude is 0.064, and for the baseline case is 0.075. This decrease of the amplitude was expected, due to its high dependence on the vortex shedding size. In addition, it can be observed how the C_l values are considerably higher for the optimum configuration than in the baseline case. A similar behaviour is observed in Fig. 7.12, with the exact same decrease in the amplitude of the fluctuations. In this case, the amplitude for the optimum configuration is 0.0168, and for the baseline case is 0.0225. It can also be observed how the C_d values for the optimum configuration are lower than those obtained in the baseline case.

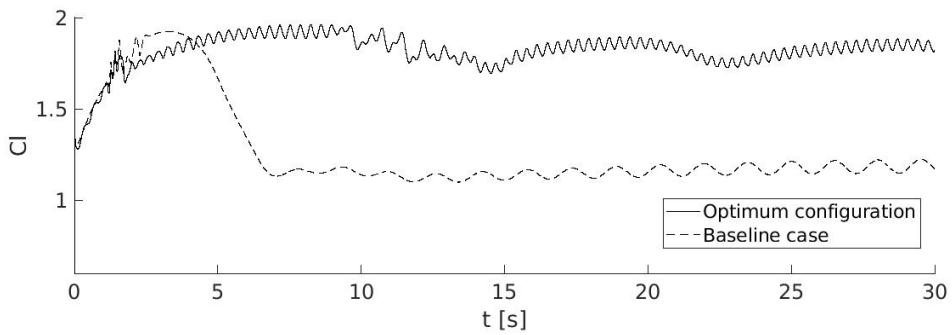


FIGURE (7.11) C_l VS time for the optimum configuration and the baseline case

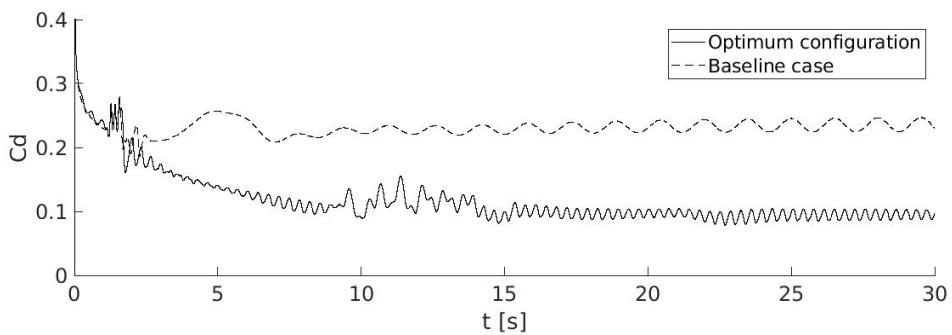


FIGURE (7.12) C_d VS time for the optimum configuration and the baseline case

A comparison of the velocity fields for the optimum configuration and the baseline case has been presented in Fig. 7.13. It can be observed how the separation point has been considerably displaced in respect with the baseline case, confirming that the boundary layer has been successfully reattached with the implementation of

AFC. In addition, the appearance of small bubbles, corresponding to the moving vortex, can be observed in extrados for the optimum configuration.

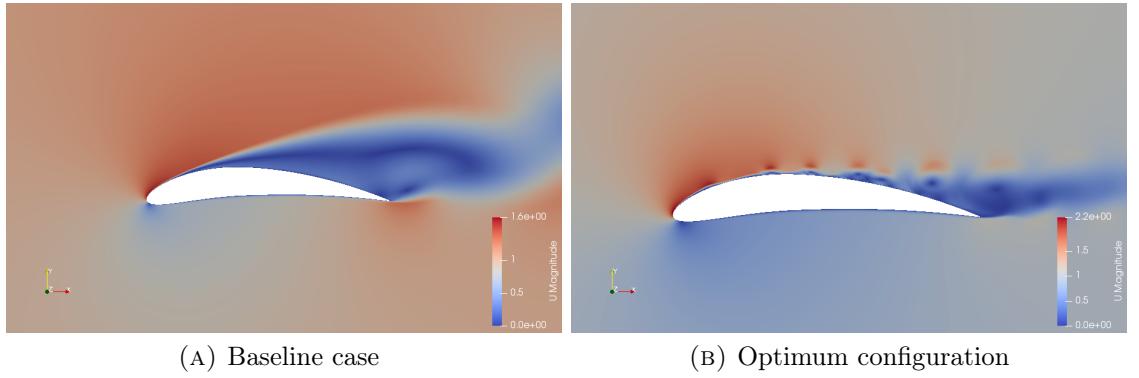


FIGURE (7.13) Velocity field comparison

In Fig. 7.14, a comparison of the pressure contours obtained for the baseline case and the optimum configuration is presented. In the first image, corresponding to the baseline case, the vortex can be observed over extrados, and its size can be noted in the wake region. In comparison, in the second image, 5 vortex of smaller dimensions can be observed over extrados. In addition, observing the wake region, it can be stated that the vortex shedding frequency has been increased.

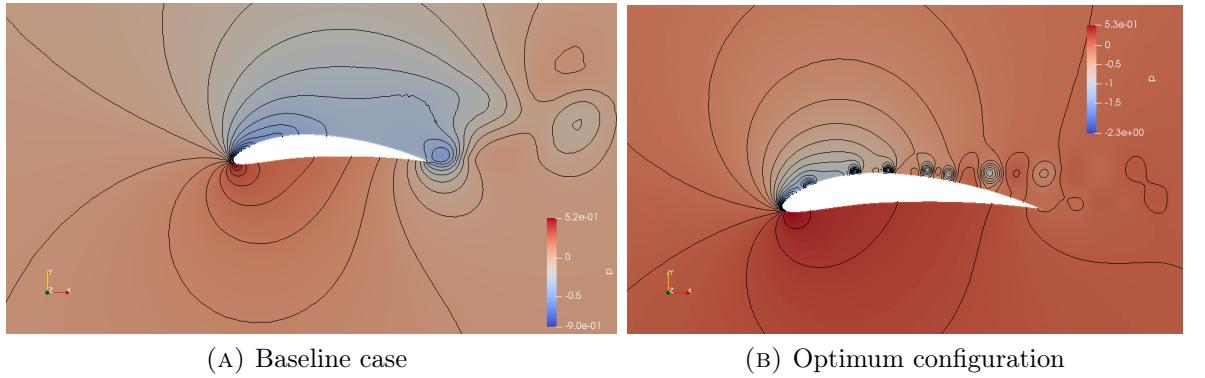


FIGURE (7.14) Pressure contour comparison

Both Fig. 7.13 and Fig. 7.14 have been obtained from videos of the optimum case created with ParaView, which have helped to extract conclusions about the flow behaviour. Videos showing the flow behaviour for different cases will be shown during the oral presentation.

Chapter 8

Conclusions

The automation of the mesh generation has shown a considerable time saving when designing the necessary meshes to fulfill the study. As a result, this feature of the project has been very useful to generate meshes with a decent level of accuracy, enabling to focus on their quality with a higher level of detail. However, it has been detected that the wake region had an excessive accuracy level, being translated in computational time that could be saved.

The Mesh Independence Test has shown a decreasing trend in the relative error between mesh candidates, which helped to visualize the level of accuracy of each mesh. In addition, the increasing accuracy with lower y^+ values has been clearly observed in the temporal representation of C_l and C_d . With lower y^+ values, the fluctuations of C_d and C_l have become greater, indicating that the mesh is accurate enough to capture the boundary layer. Even though the mesh with $y^+=0.3$ showed the best results, $y^+=0.5$ has been selected to study the baseline case. This decision has been motivated by the high computational demand that $y^+=0.3$ presented, which required approximately 7 days to simulate 30 computational seconds. Taking into account that this Thesis had to be developed in 4 months, this simplification has been considered reasonable given the level of detail proper of a bachelor degree thesis. In addition, the difference of C_l , C_d , C_P and C_F between $y^+=0.3$ and $y^+=0.5$ has been considered small enough to consider $y^+=0.5$ as a proper mesh candidate.

Regarding the baseline case results, the behaviour shown is in accordance with the solution expected. Given the high AoA of the case, the separation point is located near the LE. In addition, a vortex of big dimensions appears in extrados as a result of the flow separation, as the C_F plot has indicated. The C_P distribution is in accordance with the theory too, presenting a suction peak very close to the LE, and a progressive increase of the negative C_P values with x/C . The convergence study has indicated that the results needed more computational time in order to be statistically fully reliable. However, due to the computational requirements that would suppose increasing the computational time for each simulation, the results have been considered correct with the acceptance of the Thesis' director.

All the AFC cases simulated have shown an increase of the aerodynamic efficiency of the airfoil in respect with the baseline case. For the optimum θ and C_μ configuration, the aerodynamic efficiency obtained has been $E=12.331$, which supposes an increase of E of the 140.53% in respect with the baseline case. The introduction of forcing frequencies has resulted in an improvement of the aerodynamic efficiency for every $F^+ < 1$. However $F^+=0.5$ showed the lowest aerodynamic efficiency of all the AFC cases studied. The optimum case has been obtained for $F^+=4$, with $E=19.2649$. This aerodynamic efficiency supposes an increment of the 275,78% in respect with the baseline case.

The implementation of AFC has shown how the huge vortex generated in extrados for the baseline case is no longer present, and instead smaller vortex moving downstream can be observed. Consequently, the vortex shedding has been successfully modified and its improvement on aerodynamic efficiency has been confirmed. The separation point has moved downstream, indicating that the boundary layer has been reattached. In addition, each AFC case studied has presented a lowest C_d value than the baseline case, as well as higher C_l values. All things considered, the objective of the study has been successfully achieved, obtaining a significant improvement of the aerodynamic efficiency by implementing AFC.

8.1 Future work

Both the solver and the turbulence model should be validated in order to evaluate if the ones used are the best-fitting option for the studied case. Thus, a comparison between solvers and turbulence models using already existing studies, or by simulating the baseline case with the different options, should be carried out.

More meshes should be designed with the objective of ensuring that the results are completely independent of the mesh. It should be noted that, in order to accomplish it, more computational power would be necessary, as the time required to carry out the MIT would increase considerably.

The residuals obtained have indicated that the results are not fully statistically reliable. As a result, the computational time should be extended to consider the solution as converged. Furthermore, by extending the simulations it could be evaluated if there are long-term variations in the aerodynamic coefficients.

It has been mentioned in multiple times that the jet width and its position in the chord also influence the modification of the boundary layer. For this reason, not only the influence of C_μ , θ and $F+$ should be studied, but also the two parameters mentioned. By doing it, the AFC implementation would consider all the design variables affecting this technology.

Finally, during Chap. 7 it has been observed how, for certain θ cases, a clear peak on C_l , C_d and E couldn't be obtained. This indicates that the C_μ range selected contemplates too few cases to obtain the most efficient configuration for the three parameters studied.

Bibliography

- [1] Anderson Jr, J. D., Fundamental of Aerodynamics. McGraw-Hill, 5th ed., 1984.
- [2] “Structure of turbulent boundary layers.” [https://geo.libretexts.org/Bookshelves/Sedimentology/Book%3A_Introduction_to_Fluid_Motions_and_Sediment_Transport_\(Southard\)/04%3A_Flow_in_Channels/4.05%3A_Structure_of_Turbulent_Boundary_Layers](https://geo.libretexts.org/Bookshelves/Sedimentology/Book%3A_Introduction_to_Fluid_Motions_and_Sediment_Transport_(Southard)/04%3A_Flow_in_Channels/4.05%3A_Structure_of_Turbulent_Boundary_Layers). [Accessed on 02/06/2020].
- [3] Cattafesta, L. and Sheplak, M., “Actuators for active flow control,” Annual Review of Fluid Mechanics, pp. 247–272, Jan. 2011.
- [4] Montazer, E., “Optimization of a synthetic jet actuator for flow control around an airfoil,” IOP Conference Series: Materials Science and Engineering, vol. 152, 2016.
- [5] Hirsch, C., Numerical Computation of Internal External Flows, vol. 1. Elsevier, 2nd ed., 2007.
- [6] Braslow, A. L., A History of Suction-Type Laminar-Flow Control with Emphasis on Flight Research, vol. 13. NASA History Division, 1999.
- [7] Salim, M. S. and Cheah, S., “Wall $y+$ strategy for dealing with wall.bounded turbulent flows,” International MultiConference of Engineers and Computer Scientists, vol. 2, Mar. 2009.
- [8] “The finite differences method.” https://www.1j11.math.upmc.fr/frey/cours/UdC/ma691/ma691_ch6.pdf. [Accessed on 10/06/2020].
- [9] “Finite volume.” https://www.cfd-online.com/Wiki/Finite_volume. [Accessed on 10/06/2020].

- [10] “Finite element.” https://www.cfd-online.com/Wiki/Finite_element. [Accessed on 12/06/2020].
- [11] “Simplefoam.” <https://openfoamwiki.net/index.php/SimpleFoam>. [Accessed on 26/03/2020].
- [12] “Pisofoam.” <https://openfoamwiki.net/index.php/PISO>. [Accessed on 28/03/2020].
- [13] “Which turbulence model should i choose for my cfd application?” <https://www.comsol.com/blogs/which-turbulence-model-should-choose-cfd-application/>. [Accessed on 10/04/2020].
- [14] “k-epsilon.” <https://www.simscale.com/docs/simulation-setup/global-settings/k-epsilon/>. [Accessed on 01/05/2020].
- [15] Batikh, A., Baldas, L., and Colin, S., “Application of Active Flow Control on Aircrafts - State of the Art,” International Workshop on Aircraft System Technologies, Feb. 2017.
- [16] De Giorgi, M. G., De Luca, C. G., Ficarella, A., and Marra, F., “Comparison between synthetic jets and continuous jets for active flow control: Application on a NACA 0015 and a compressor stator cascade,” Aerospace Science and Technology, vol. 43, pp. 256–280, Aug. 2015.
- [17] Crowther, W. J. and Jabbal, M., “Flow control fallacies: a review of common pitfalls in flow control research,” Proc. Inst. Mech Eng. Part G J. Aerosp. Eng, vol. 225, p. 1–11, Jan. 2011.
- [18] Lengers, M., “Industrial assessment of overall aircraft driven local active flow control,” Proceedings of the 29th Congress of the International Council of the Aeronautical Sciences, pp. 1–11, Sept. 2014.
- [19] Bauer, M., “Wing tip drag reduction at nominal take-off mach number: An approach to local active flow control with a highly robust actuator system,” MDPI, Oct. 2016.

- [20] “Gmsh. a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities.” <https://gmsh.info>. [Accessed on 20/06/2020].
- [21] “Paraview.” <https://www.paraview.org/>. [Accessed on 20/06/2020].
- [22] “Naca 4 digit airfoil generator.” <http://airfoiltools.com/airfoil/naca4digit>. [Accessed on 25/02/2020].
- [23] Anderson Jr, J. D., Computational fluid dynamics: The basics with applications. McGraw-Hill, 1995.
- [24] “How can i understand mesh quality and simulation results in cfd.” https://www.researchgate.net/post/How_can_I_understand_mesh_quality_and_simulation_results_in_CFD. [Accessed on 24/03/2020].
- [25] “Mesh quality.” <https://www.afs.enea.it/project/neptunius/docs/fluent/html/ug/node167.htm>. [Accessed on 24/03/2020].
- [26] Tousi, N. M., Bergadà, J. M., and Rodriguez, I., “Active flow control on SD7003 airfoil using synthetic jets,” 5th International Conference on Advances in Mechanical Engineering Istanbul 2019, Dec. 2019.
- [27] Catalano, P. and Tognaccini, R., “Turbulence modeling for low-reynolds-number flows,” AIAA Journal, vol. 48(8), pp. 1673–1685, 2010.
- [28] “Standard boundary conditions.” <https://www.openfoam.com/documentation/user-guide/standard-boundaryconditions.php>. [Accessed on 20/04/2020].
- [29] “Openfoam v6 user guide: 5.1 mesh description.” <https://cfd.direct/openfoam/user-guide/v6-mesh-description/>. [Accessed on 10/03/2020].
- [30] “Decomposepar.” <https://openfoamwiki.net/index.php/DecomposePar>. [Accessed on 10/05/2020].
- [31] Minelli, G., Krajnovic, S., and Basara, B., “Numerical Investigation of Active Flow Control Around a Generic Truck A-Pillar,” Flow, Turbulence and Combustion, vol. 97, pp. 1235–1254, Aug. 2016.

- [32] Grund, T. and Nietsche, W., “Active flow control on a s10 glider configuration,” Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Apr. 2010.
- [33] Jané, C., “Study of flow control mechanisms with openFoam,” Universitat Politècnica de Catalunya, Aug. 2019.
- [34] Tuck, A. and Soria, J., “Active Flow Control over a NACA 0015 airfoil using a ZNMF jet,” 15th Australasian fluid mechanics conference, pp. 13–17, 1991.
- [35] Cierpka, C., Weier, T., and Gerbeth, G., “Electromagnetic control of separated flows using periodic excitation with different wave forms,” Active Flow Control. Papers contributed to the Conference “Active Flow Control 2006”, vol. 95, pp. 27–41, Sep. 2006.