

Technische Universität Kaiserslautern

Fachbereich Maschinenbau und Verfahrenstechnik

Strömungsmechanik, Akustik und Strömungsmaschinen

Diplomarbeit

**AUTOMATIC OPTIMIZATION
OF AN AIRFOIL FOR
OCEAN CURRENT TURBINES**

by

Víctor Rodríguez Rodríguez

Tutor: **Dr. Harald Roelawski**

October 2012

CONTENTS

CONTENTS

ABSTRACT.....	6
1. INTRODUCTION.....	8
1.1. Background.....	8
1.1.1. <i>Ocean Energy</i>	8
1.1.2. <i>Technology</i>	16
1.2. Aims/Objectives.....	17
2. METHODOLOGY.....	20
2.1. Problem description.....	20
2.2. Airfoil geometry generation on C++.....	23
2.3. Geometry on MATLAB.....	24
2.4. Flow simulation on ANSYS Workbench.....	24
2.4.1. <i>Geometry</i>	24
2.4.2. <i>Mesh</i>	25
2.4.3. <i>Problem setup</i>	27
2.4.4. <i>Calculation and post-processing</i>	27
2.5. MATLAB commanding role and optimization.....	30
3. RESULTS AND DISCUSSION.....	37

4. CONCLUSION AND FURTHER RESEARCHES.....	47
5. REFERENCES.....	51
APPENDIX.....	53

ABSTRACT

ABSTRACT

In this study the kind of airfoil used is a variant from the commonly utilized ones, due to the conditions in which it will have to work. Instead of the usual applications where the fluid is air, in this case the airfoil must operate in water.

The aim within the whole work is to optimize the airfoil shape. To be precise, optimizing means maximizing the aerodynamic force called lift on the airfoil. The lift to drag ratio will be another important control parameter during all the process. There is a specific shape given some certain boundary conditions and constraints that makes possible to obtain the maximum value of the lift.

What has been done is, in short, starting from a C++ code already developed in charge of the airfoil geometry generation, adapt it so as to be able to manually introduce variations on the geometry. The geometry will be imported on ANSYS Workbench, properly meshed afterwards and then the problem will be solved on Fluent.

The final step, i.e., the optimization, will be carried out by the Optimization Toolbox included in MATLAB. MATLAB is also the tool that commands all the process, calling subroutines and launching other programs from its DOS command.

1

INTRODUCTION

1. INTRODUCTION

1.1. BACKGROUND

Nowadays, all the applications where airfoils are used are innumerable. To begin with, probably the most visual one: the wings in aircrafts. In the same way, the energy production applications must also be taken into account (wind or water turbines, combined cycle power plants, etc.).

1.1.1. Ocean Energy

There is currently a renewed interest in using the ocean to generate electricity, using both traditional hydropower technologies and new hydrokinetic technologies. This interest is promoting development of cleaner energy sources and diversification of energy supplies through use of alternative and renewable sources.

Turbines (analogous to the wind power) use in this case the kinetic energy from a working fluid (in this case water from ocean currents, both tidal and permanent currents) and transform it into electrical power. Because of their similar physical properties, ocean current turbines are similar to wind power turbines. Sea water, whose density is 832 times bigger than air density, gives a 5 knots (2,5 km/h) ocean current more kinetic energy than a 350 km/h wind; therefore ocean currents have a very high energy density. Hence a smaller device is required to harness ocean current energy than to harness wind energy. The

relationship between the power from the water current, water density and velocity is given by $P = \frac{1}{2}\rho AV^3$.

For applications in tidal currents, where the direction of flow is changed periodically (normally every about 6 hours), rotor blades are used to operate from both sides. Accordingly the turbine changes the direction of rotation at each tidal change. [1]

This renewable energy source technology is applicable wherever constant strong currents exist within close proximity to land and populations. Some of these areas include southeastern Florida, Caribbean Islands and Japan. This continuous renewable energy source is designed to have virtually no environmental impacts.

The main advantages of the use of ocean current energy are:

- The first one, and already commented, the high energy density, makes ocean energy be ideal for large-scale developments in a multiple power range.
- The high load factors derived from the fluid properties. The predictability of the resource, so that, unlike most of other renewables, the future availability of energy can be known and planned for.
- The potentially large resource that can be exploited with little environmental impact, thereby offering one of the least damaging methods for large-scale electricity generation. [5] In addition to this lack

of environmental impact, it can also be said that as they are installed beneath the water surface, these turbines have a minimal visual impact.

- Ocean energy projects also may be useful in filling in the gaps in generation from other intermittent energy sources, such as wind projects.[6]

- Types and causes of ocean currents

The term ocean currents as used here includes any kind of water currents, be it tidal currents, or other ocean currents driven for instance by thermal gradients or differences in salinity. In addition to their varying size and strength, ocean currents differ in type. They can be either surface or deep water.

Surface currents are those found in the upper 400 meters of the ocean and make up about 10% of all the water in the ocean. Surface currents are mostly caused by the wind because it creates friction as it moves over the water. This friction then forces the water to move in a spiral pattern, creating gyres. In the northern hemisphere, the wind spins clockwise and in the southern, anticlockwise. The speed of surface currents is greatest closer to the ocean's surface and decreases at about 100 meters below the surface. Because surface currents travel over long distances, the Coriolis Force also plays a role in their movement and deflects them, also being responsible for the creation of their circular pattern. If the earth did not rotate, wind would travel the globe in straight lines. Instead, the spin of the earth causes winds to seemingly curve to the right in the Northern Hemisphere and the left in the Southern Hemisphere. This curvature of the winds is known as the Coriolis Effect.

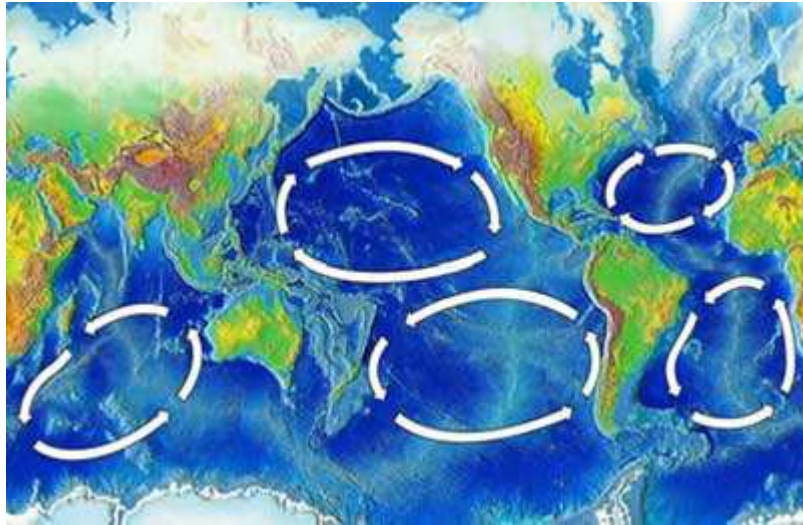


Figure 1.1: Circular wind patterns derived from the Coriolis Force

(Source: <http://science.howstuffworks.com/environmental/earth/oceanography/ocean-current2.htm>)

Deep water currents, also called thermohaline circulation, are found below 400 meters and make up about 90% of the ocean. Like surface currents, gravity plays a role in the creation of deep water currents but these are mainly caused by density differences in the water. Density differences are a function of temperature and salinity. Warm water holds less salt than cold water so it is less dense and rises toward the surface while cold, salt laden water sinks. As the warm water rises though, the cold water is forced to rise through upwelling and fill the void left by the warm. By contrast, when cold water rises, it too leaves a void and the rising warm water is then forced to descend and fill this empty space, creating thermohaline circulation. Thermohaline circulation is known as the “Global Conveyor Belt” because its circulation of warm and cold water acts as a submarine river and moves water throughout the ocean. [7]

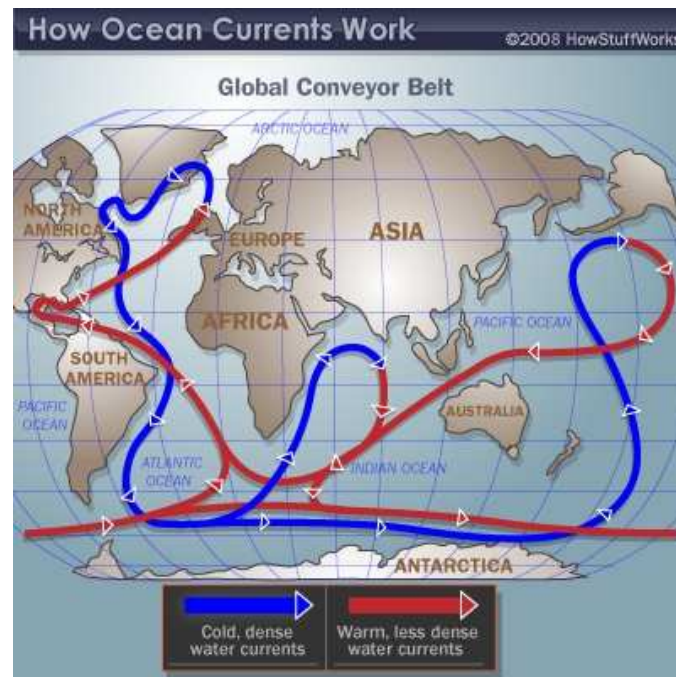


Figure 1.2: Global Conveyor Belt

(Source: <http://science.howstuffworks.com/environmental/earth/oceanography/ocean-current3.htm>)

Tidal currents, as their name suggests, are created by tides, caused by the gravitational forces from the moon, and in less influence from the sun.



Figure 1.3: effect of the gravitational forces from the Moon and the Sun

(Source: <http://science.howstuffworks.com/environmental/earth/oceanography/ocean-current4.htm>)

This rise in water level is accompanied by a horizontal movement of water called the tidal current. The gravitational pull of the moon usually creates two high tides and two low tides each day. Although tides and tidal currents don't have much impact in the open oceans, they can create a rapid current of up to 25 km/h when they flow in and out of narrower areas like bays, estuaries and harbors.

The strongest tidal currents occur at or around the peak of high and low tides. When the tide is rising and the flow of the current is directed towards the shore, the tidal current is called the flood current, and when the tide is receding and the current is directed back out to sea, it is called the ebb current. Because the relative positions of the moon, sun and earth change at a known rate, tidal currents are predictable.

- HOW GREAT is this potential?

The total worldwide power in ocean currents has been estimated to be about 5,000 GW, with power densities of up to 15 kW/m². The relatively constant extractable energy density near the surface of the Florida Straits Current is about 1 kW/m² of flow area. It has been estimated that capturing just the 0,1% of the available energy from the Gulf Stream, which has 21000 times more energy than Niagara Falls in a flow of water that is 50 times the total flow of all the world's freshwater rivers, would supply Florida with 35% of its electrical needs. [3] To get an overview in a higher level, both the U.S. and the U.K., for example, have enough ocean power potential to meet around 15% of their total power needs.

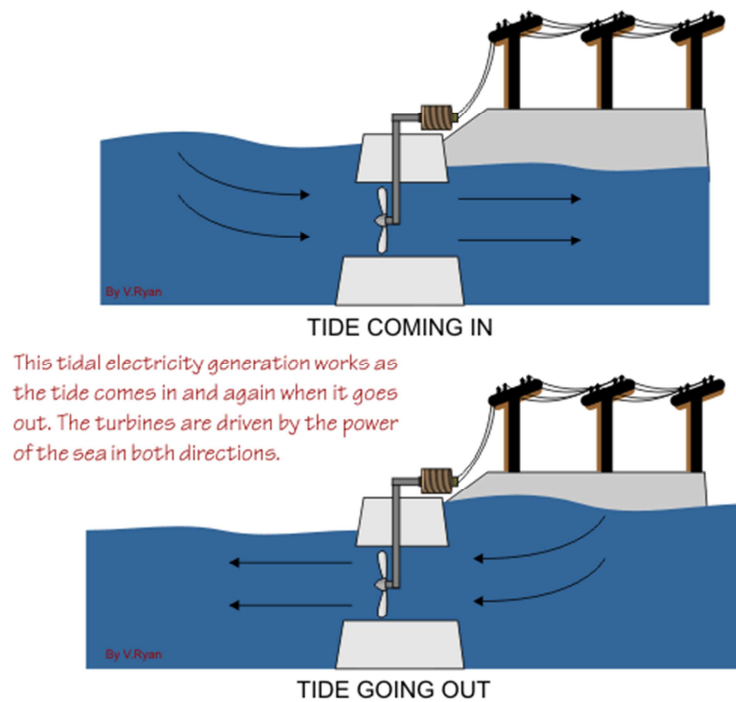


Figure 1.4: operation with tidal currents

(source:<http://www.alternative-energy-news.info/technology/hydro/tidal-power/>)

- WHERE is this potential mainly located?

Areas that typically experience high marine current flows are in narrow straits, between islands and around headlands. Entrances to lochs, bays and large harbors often also have high marine current flows. Generally the resource is largest where the water depth is relatively shallow and a good tidal range exists. In particular, large marine current flows exist where there is a significant phase difference between the tides that flow on either side of large islands. [4]

There are many zones in the world with current velocities of 2,5 m/s and greater. Countries with an exceptionally high resource include the UK, Ireland, Italy, the Philippines, Japan and parts of the United States. A few studies have been carried out to determine the total global marine current resource, although it is estimated to exceed 450 GW. [4]

In the United States, the Florida Current and the Gulf Stream are reasonably swift and continuous currents moving close to shore in areas where there is a demand for power. If ocean currents are developed as energy sources, these currents are among the most likely. But most of the wind-driven oceanic currents generally move too slowly and are found too far from where the power is needed. Here is a map of all major known world-wide ocean currents.

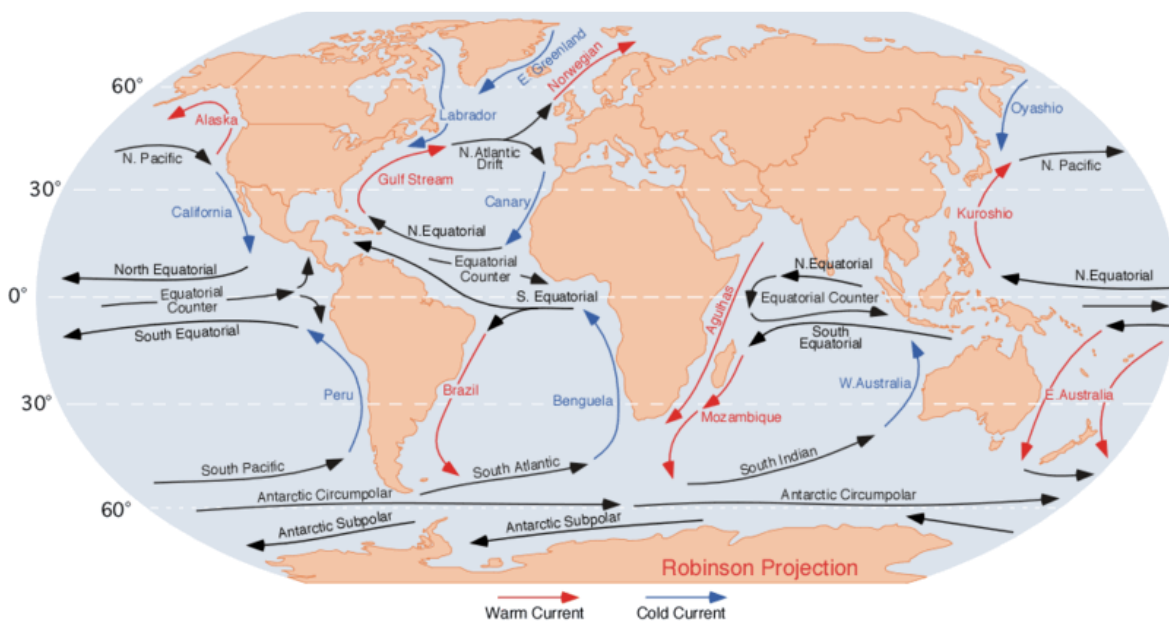


Figure 1.5: main ocean currents

(Source: <http://blue.utb.edu/paullgj/geog3333/lectures/physgeog.html>)

1.1.2. Technology

Technology involves submerged turbines, anchored to the ocean floor. Energy can be extracted from the ocean currents using submerged turbines that are similar in function to wind turbines, capturing energy through the processes of hydrodynamic lift or drag. These turbines would have rotor blades, a generator for converting the rotational energy into electricity, and a means for transporting the electrical current to shore for incorporation into the electrical grid.

Turbines can have either horizontal or vertical axes of rotation. Mechanisms such as posts, cables, or anchors are required to keep the turbines stationary relative to the currents with which they interact.

In large areas with powerful currents, it is possible to install water turbines in groups or clusters to create a “marine current facility,” similar to wind turbine facilities. Turbine spacing would be determined based on wake interactions and maintenance.

For marine current energy to be utilized, a number of potential problems need to be identified, including avoidance of drag from cavitation, prevention of marine growth accumulation, corrosion control, and system reliability (especially important due to the maintenance complexity and its potentially high costs).

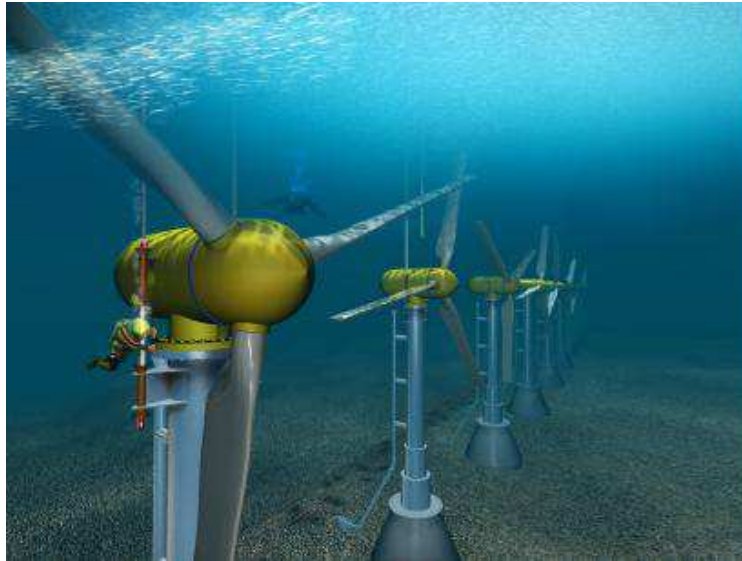


Figure 1.6: underwater turbines

(Source: <http://ocsenergy.anl.gov/guide/current/index.cfm>)

1.2. AIMS/OBJECTIVES

In this work, as it has already been mentioned, the main objective is to find an optimal design of an airfoil shape used for ocean current energy applications, so that this airfoil design can generate the maximum hydrodynamic lift. A maximum value of lift means an optimized value of the moment generated on the rotor, and its last consequence, a maximum value of the power generated with the device. A multipurpose system can also be defined, in order to minimize the other hydrodynamic force on the blades, i.e. the drag.

To attain this global aim, some other partial objectives must also be identified:

- To adapt an existing source code in C++ programming language that defines all the airfoil geometry, so that the input variables may manually be defined.

- Using the different airfoil geometries defined with the C++ code, to be able to import this geometry in ANSYS Workbench, which by using one of its CFD tools (Fluent) will numerically simulate the problem and display the values of the lift and the drag on the airfoil in every case.

- To manage all the process and workflow, some MATLAB scripts and functions will be developed. Through them, the input variables can be manually defined, the airfoil geometry updated in every case, the geometry imported in the CFD software and the values of lift and drag extracted after every simulation.

- To be able to handle an optimization tool, via MATLAB. In addition to its control role, MATLAB will be in charge of this last step, through its Optimization Toolbox. This toolbox, by using different algorithms and criteria, manages to find the optimal of a specific function within some determined constraints and boundary conditions.

2

METHODOLOGY

2. METHODOLOGY

To give an overview of the work, a flowchart is firstly developed:

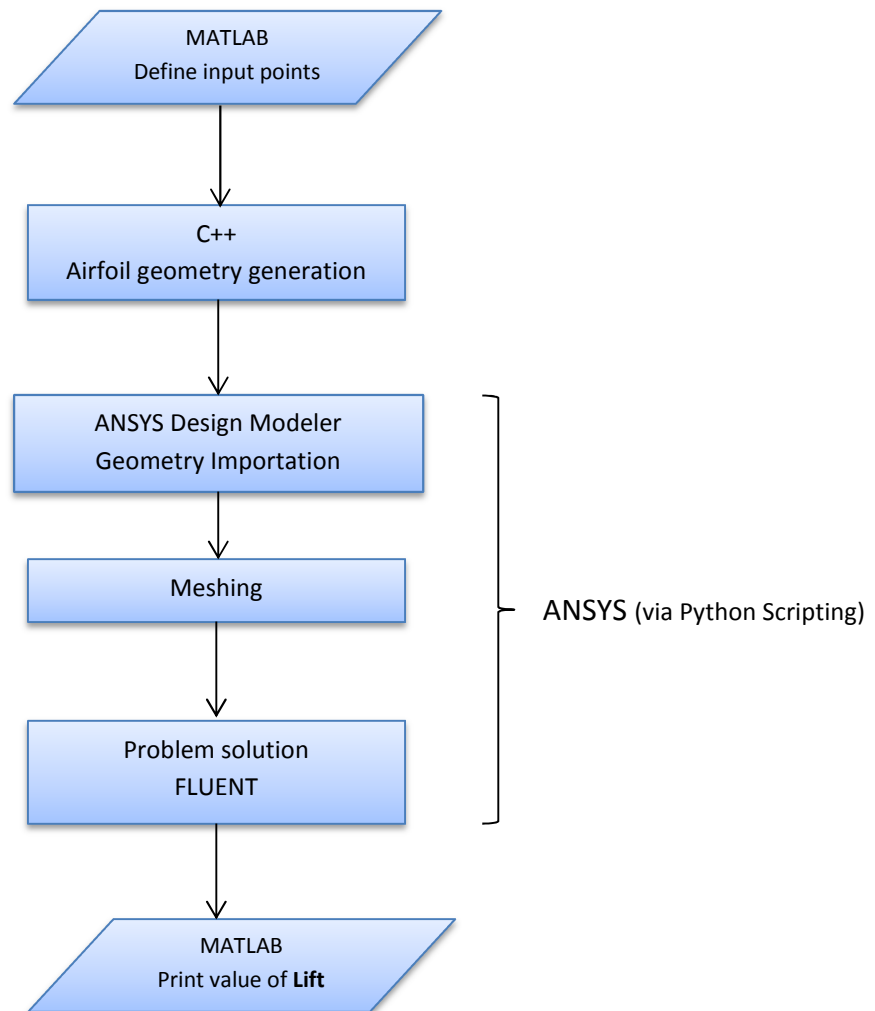


Figure 2.1: Flowchart describing the whole workflow

2.1. PROBLEM DESCRIPTION

The special feature of this study is a propeller turbine that may operate from both sides, i.e. perform cyclic reversals (typical in tidal currents). Why is this

configuration advantageous? If the flow reverses (for example the change of the tidal flow), the performance of a conventional profile would become seriously poor, i.e. the values of the lift and lift to drag ratio would be drastically reduced. The current would flow from the trailing edge side, what would lead to extremely big losses, since the flow would not adapt properly to the airfoil surface. This is the reason why conventional profiles are designed to operate from the rounded leading edge (see figure 2.2).

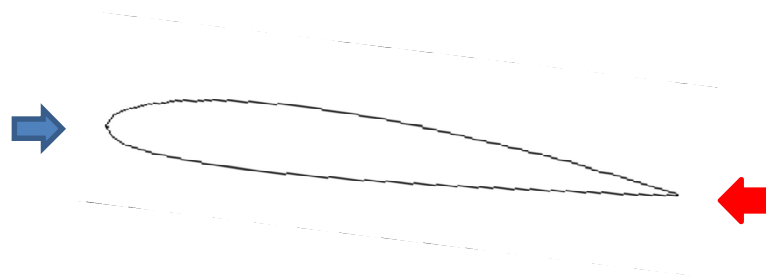


Figure 2.2: conventional profile unable to perform a current flow from the right

In the new design (see figure 2.3), whose most important feature is its symmetry, both edges can act as leading edge or trailing edge, in accordance with the current flow direction. Thus, the flow direction limitation is avoided, and the objective is improving its design so that it may operate with the best possible performance.

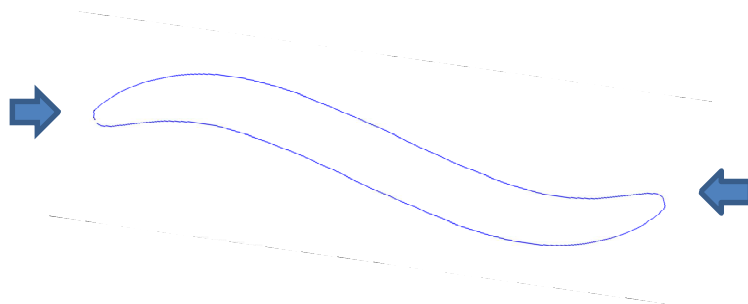


Figure 2.3: profile able to perform changing current flows

The profile (figure 2.4) is formed by a camber line and a thickness distribution. The camber line is based on a Bézier spline. The start and end point are both fixed to the spline. Two control points define the curvature of the spline, and their XY coordinates are the variables of the problem. Each different pair of these points defines a different geometry. These two points are the first half of the geometry, whilst their two antisymmetric points are placed at the second half.

The thickness distribution is based on the Bisuper Ellipse equation:

$$\left(\frac{x}{l}\right)^{m_1} + \left(\frac{y}{d}\right)^{m_2} = 1$$

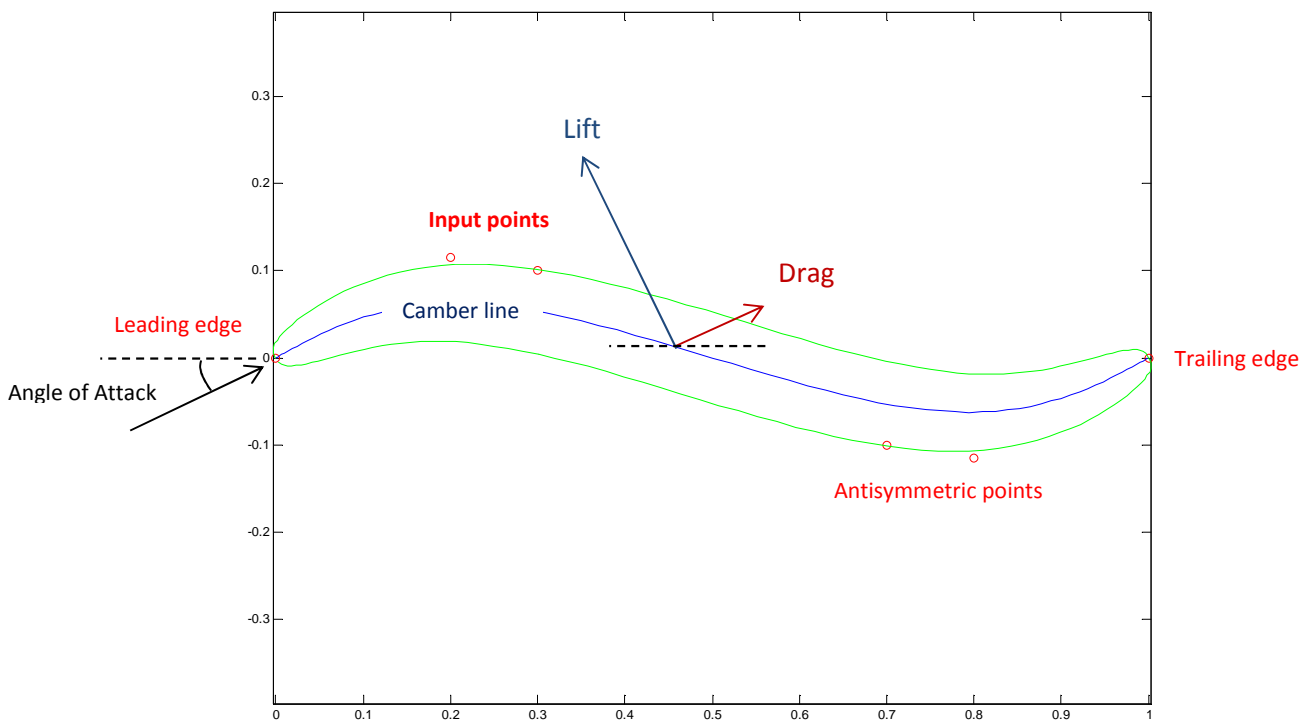


Figure 2.4: profile and features

Note that, in this case, the whole geometry is nondimensional, hence the scale between 0 and 1. Later on, the variable corresponding to the length will be added.

Consequently, the aim of the whole problem is to find the combination of variables (geometry points plus length) that leads to a maximum value of the lift, given certain boundary conditions.

2.2. AIRFOIL GEOMETRY GENERATION ON C++

Starting from a previously developed C++ code, which generated the geometry given the coordinates of the two points that define the airfoil shape (using the Bézier spline and the Bisuper ellipse, as previously described in the section 2.1), the first task was modifying the code, so that the points' coordinates were available as the input variables of the problem.

The initial C++ code was such that the points could not be defined from the command prompt, but they were implicitly defined in the code. The first thing to be done was changing the kind of C++ function, from `void(main)`, which is a kind of function that does not accept input or print any output, to `int main(int argc, char* argv[])`. Now that an integer function that accepts inputs is defined, these inputs are now callable from the command prompt by using the command "*atof*", that transfers the value of every input argument to the variables (`x[1]=atof(argv[1])`).

Finally, after debugging the new code, the resulting executable file was created.

2.3. GEOMETRY ON MATLAB

Launching the executable coming from the C++, by using the DOS command on MATLAB, a data file containing the coordinates of all the airfoil points is created, so now the profile can be plotted on MATLAB (as shown in the figure 2.4). The input points defining the shape, as well as the camber line distribution, are also saved to data files and plotted.

2.4. FLOW SIMULATION ON ANSYS WORKBENCH

For this part, the ANSYS Workbench 12.1 version is used, and is in charge of calculating the lift for every different geometry case. For this numerical CFD analysis, a Fluent Analysis System is chosen. The workflow is controlled by a script written in Python Programming Language. The workflow progresses as follows:

2.4.1. *Geometry*

Using the ANSYS Design Modeler, it is possible, by choosing the text file where the airfoil points' coordinates are saved for every case, to import the geometry. A fluid domain is also created around the airfoil (see the figure below).

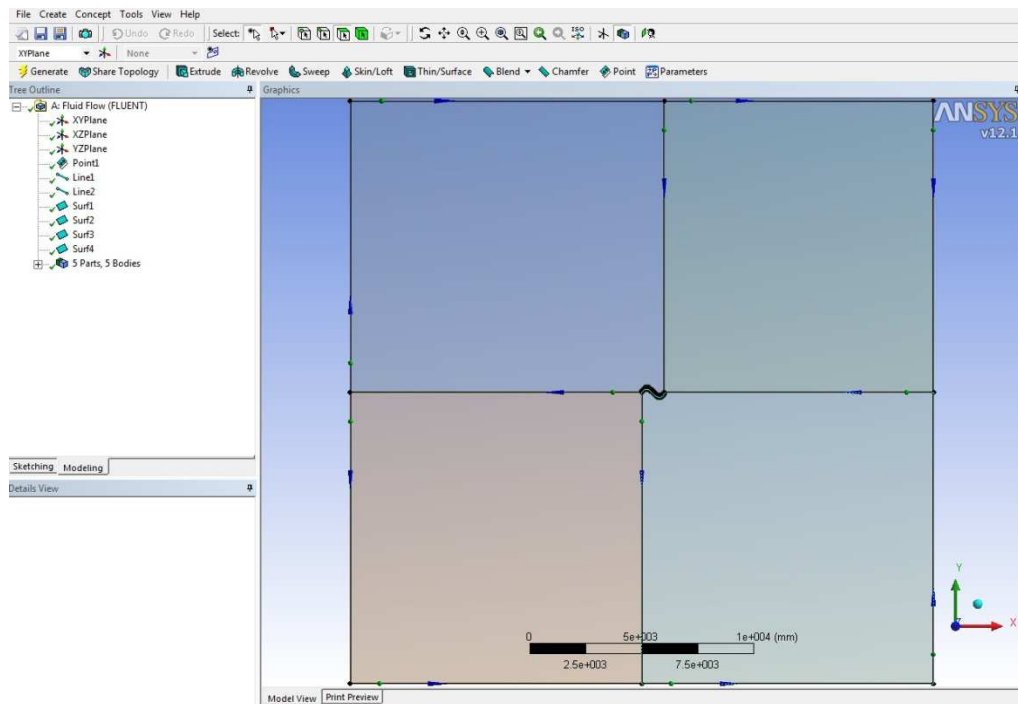


Figure 2.5: geometry definition on the Design Modeler

2.4.2. Mesh

Using the ANSYS meshing tool. The geometry from the Design Modeler is meshed, by following certain criteria, such as the number of divisions per edge. The aim is a smaller elements size around the airfoil, so that the mesh is there refined. A preliminary study (see sections 2.4.3 and 2.4.4) was needed, in order to determine whether the mesh is reliable or not for this model (checking the y^+ plot on the airfoil).

The figures 2.6, 2.7 and 2.8 show the whole mesh, zoomed near the leading edge and near the trailing edge respectively.

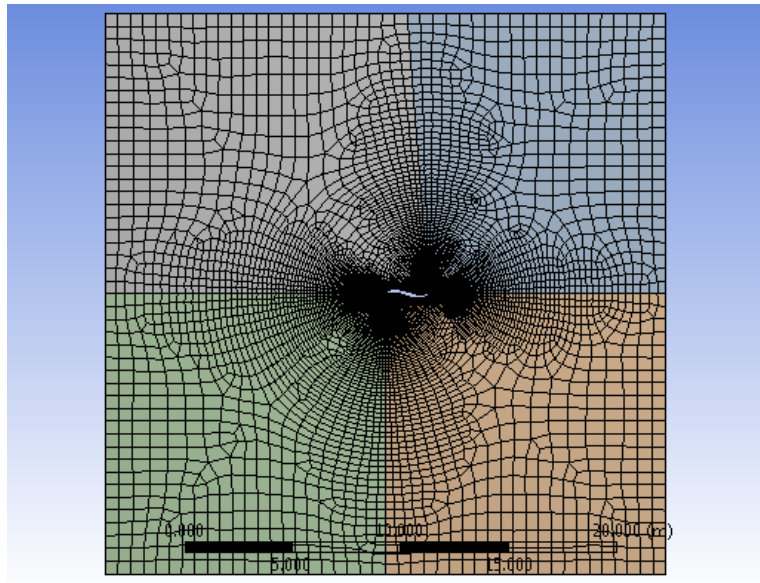


Figure 2.6: mesh

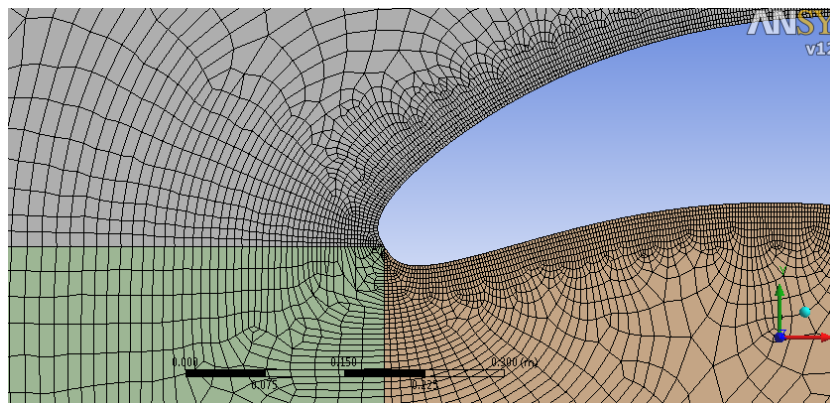


Figure 2.7: mesh near the leading edge

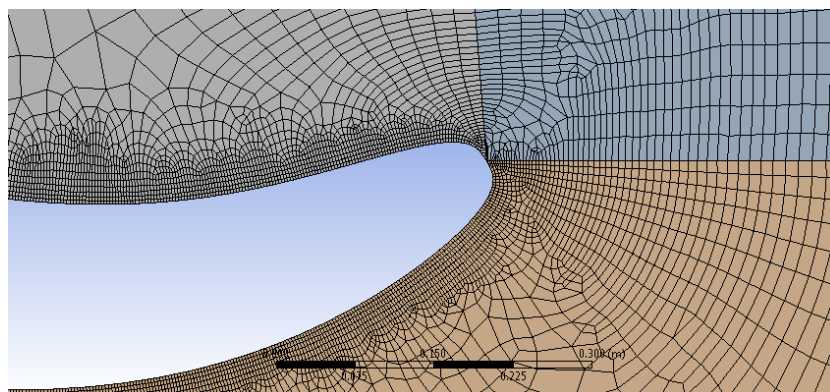


Figure 2.8: mesh near the trailing edge

2.4.3. Problem setup

The fluid flow problem is set up on *Fluent*. This is a non-compressible fluid, therefore the solver is chosen as pressure-based and the fluid is liquid water. The boundary conditions are: for the velocity inlet, the velocity is 2 m/s, the angle of attack is 5° ; for the pressure outlet, the gauge pressure is 0 Pa (atmospheric pressure); the airfoil is set as a stationary wall. The Reynolds number is $Re = \frac{\rho U_\infty L}{\mu} \sim O(10^6)$

According to the order of magnitude of the Reynolds number, and since the resolving the boundary layer would result in a mesh which would be too large for the many runs, the mesh size is chosen to be on the upper allowable y^+ limit for the wall functions ($30 \leq y^+ \leq 300$). The turbulence model chosen for the optimization is the **k-epsilon**, with standard wall functions as wall treatment.

2.4.4. Calculation and post-processing:

With the first meshing, the y^+ values were not in an acceptable range, so the mesh was refined in the normal direction to the airfoil. This refinement involved a higher computational cost, and since an optimization requires many runs to be solved.

In the Table 2.1, here it is an evolution of the lift and drag values so as to be able to determine which residuals order of magnitude is reliable to run the cases.

Residuals Tolerance	Lift (N)	Drag (N)
1.00E-04	1148	498.9
1.00E-05	560.3	476
1.00E-06	686.4	481.4

Table 2.1: Results at each convergence criteria for an S-shaped profile

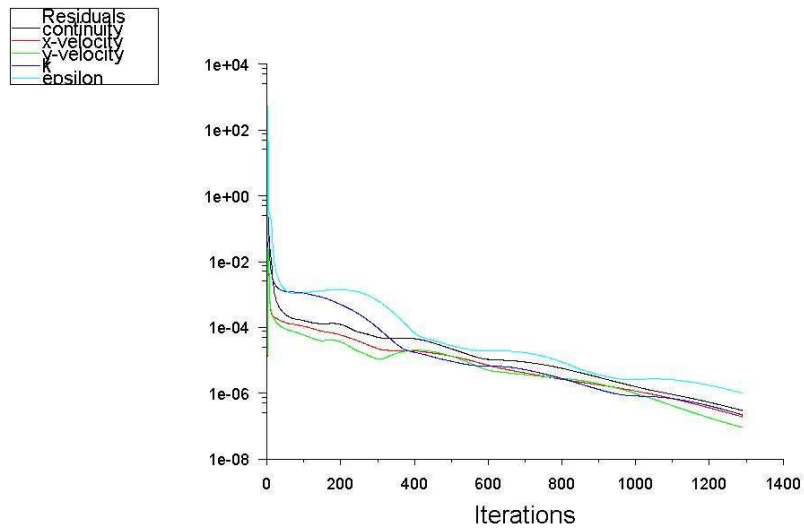


Figure 2.9: residuals convergence history for an S-shaped profile

In case of S-shaped airfoils, the computational cost is not that high even for residuals of order of magnitude 10^{-6} . Nevertheless, the situation is different when a flattened airfoil is studied.

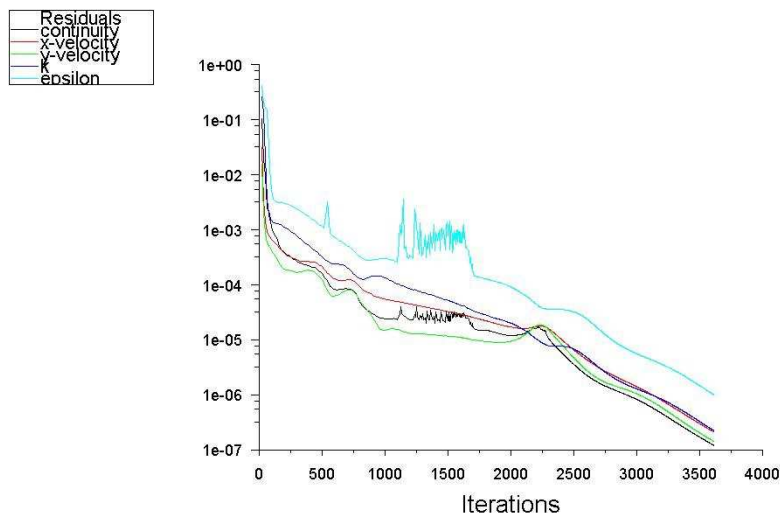


Figure 2.10: residuals convergence history for a flattened profile

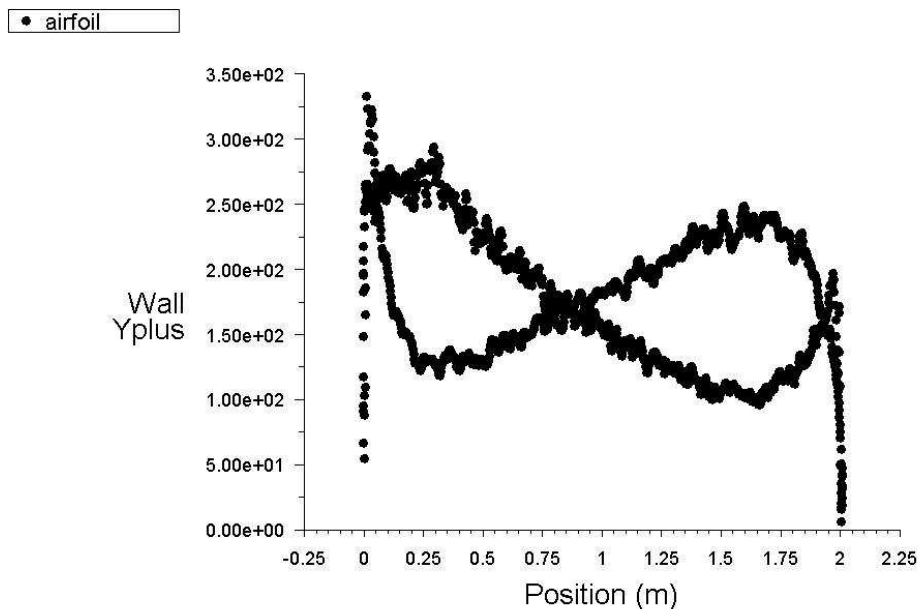
With a flat profile, the number of iterations until any of the convergence criteria is reached has been considerably increased.

Residuals Tolerance	Lift (N)	Drag (N)
1.00E-04	-915	228.9
1.00E-05	1000.7	266.8
1.00E-06	1154	271.6

Table 2.2: Results at each convergence criteria for a flattened profile

Note that for an order of magnitude of 10^{-4} , the value for the lift is negative, which means the flow field requires a greater amount of iterations to develop properly. So at least, an order of magnitude of 10^{-5} is needed. After observing the results for both geometries, since the results for 10^{-5} and 10^{-6} make no big differences, and the computational cost for a flat geometry is high, convergence criteria of residuals with order of magnitude 10^{-5} are chosen.

With the current mesh, the y^+ XY plot is:

Figure 2.11: y^+ plot on the airfoil

Since a standard wall function is used, the y^+ values acceptable range is from 30 to 300. According to the y^+ plot, the current meshed is reliable to solve the near-wall domain.

As previously commented, all the tasks on ANSYS are executed automatically, by using a Python Script, which imports and updates the geometry for every different case, the mesh and the setup on Fluent. Finally, after the calculation, the updated values of the parameters *lift* and *drag* are gathered and saved as a text file.

2.5. MATLAB COMMANDING ROLE AND OPTIMIZATION

All the steps previously described must follow a specific order. In charge of commanding these tasks, MATLAB must control the workflow. A MATLAB script is created containing all the functions and commands to this goal.

After the task of creating and plotting the airfoil geometry, MATLAB must launch the ANSYS Workbench that computes all the fluid flow numerical simulation. As this must be done automatically, the Workbench must be executed in batch mode using the DOS MATLAB command.

One of the toolboxes included in MATLAB is the *Optimization Toolbox*, which consists of several algorithms to minimize/maximize a certain function. The toolbox also includes a graphical user interface, which is possible to set up the problem with. This GUI looks as follows in the next figure

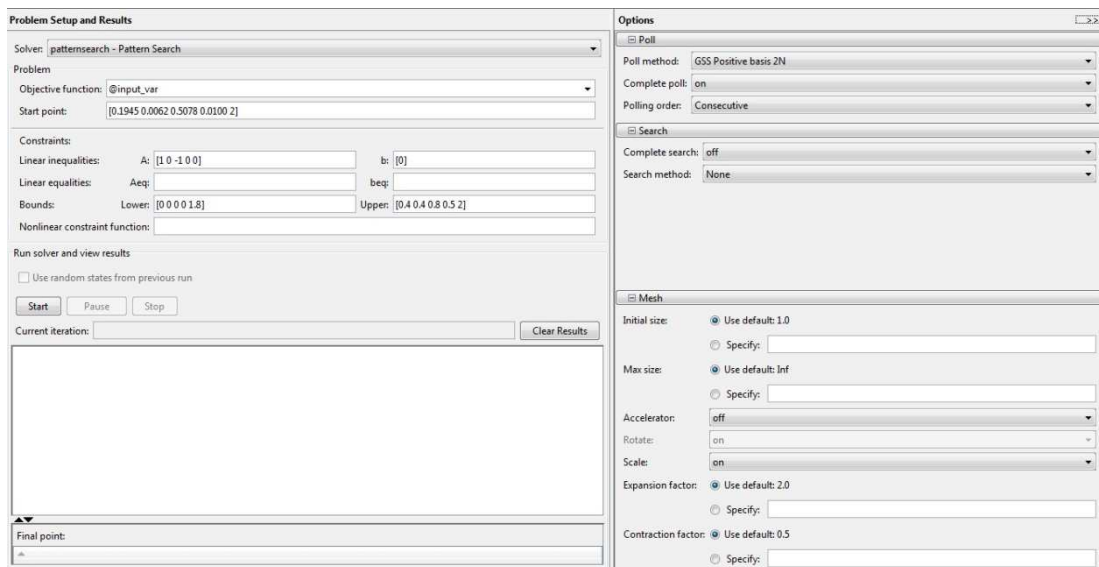


Figure 2.12: GUI of the MATLAB Optimization Toolbox

First of all, it must be decided which is the algorithm that fits the best to the sort of problem. Since there is not any mathematical expression defining the objective function (every specific case is solved and compared to the other cases), the best option is choosing the *Pattern Search* solver. Later on it will be explained how it works. The objective function to be optimized must be declared as a handle (@input_var).

Now a start point, linear equalities/inequalities and constraints must be declared. All points and constraints must be defined as arrays. The order of the variables is: x_1, y_1, x_2, y_2, L (with L airfoil length). The only linear constraint is the x -coordinate of the second design point must be bigger than the x -coordinate of the first one; i.e. $x_1 - x_2 < 0$. Bounds are considered:

$$0 \leq x_1 \leq 0.3; 0 \leq y_1 \leq 0.2; 0 \leq x_2 \leq 0.6; 0 \leq y_2 \leq 0.2; 1.9 \leq L \leq 2$$

The aim of these bounds is basically avoiding sharp and odd S-shaped profiles (figure 2.13), which would result in poorer values for lift and lift to drag ratio. Thus, the optimization will progress faster, since there are a smaller number of pattern points to be evaluated.

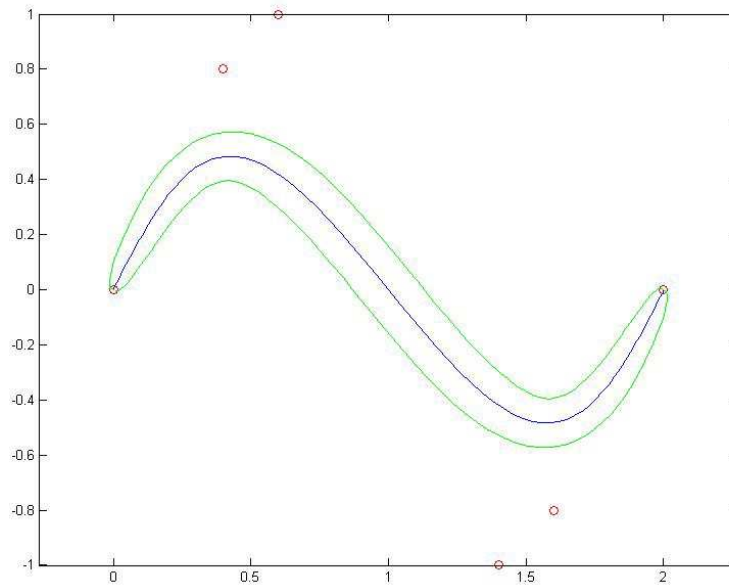


Figure 2.13: airfoil out of bounds

The *Pattern Search* solver is a method for solving optimization problems that does not require any information about the gradient of the objective function. Unlike more traditional optimization methods that use information about the gradient or higher derivatives to search for an optimal point, a pattern search algorithm searches a set of points around the current point, looking for one where the value of the objective function is lower than the value at the current point. *Pattern search* can be used to solve problems for which the objective function is not differentiable, or is not even continuous. In this case, as there is no mathematical expression for the objective function, the objective function can be considered as not continuous.

Once the function value at the current point has been calculated, the algorithm starts to calculate the value of the function at points around the current one. The poll method is called “GSS Positive Basis 2N”. This method creates a *pattern*, i.e. a set of vectors that the pattern search algorithm uses to determine which points to search at each iteration. The number of vectors depends on the number of independent variables (in this case 5). The collection of vectors that form the pattern are fixed-direction vectors. Therefore, with 5 vectors, the set is formed by the following 10 vectors:

[1 0 0 0 0]; [0 1 0 0 0]; [0 0 1 0 0]; [0 0 0 1 0]; [0 0 0 0 1];

[-1 0 0 0 0]; [0 -1 0 0 0]; [0 0 -1 0 0]; [0 0 0 -1 0]; [0 0 0 0 -1]

At each step, the algorithm polls the points in the current mesh by computing their objective function values. The algorithm multiplies the mesh size by the set of vectors, and that is how the set of points around the current point is created, then the function is evaluated once at each point. If any of these points has a bigger value of the objective function, the poll turns out successful, and the point it finds becomes the current point at the next iteration. If the algorithm fails to find a point that improves the objective function, the poll is called unsuccessful and the current point stays the same at the next iteration.

The next parameter to be taken into account is the mesh size. It is the distance between the pattern points and the current point. The coarser the mesh is, the further the poll points around the current point are. The algorithm has an expansion and contraction factor for the mesh size, which gets bigger or smaller, according to the poll. The expansion takes place when the poll is successful (a better function value has been found), and in the other hand, if it is unsuccessful (none of the polled points improves the function value), the mesh size is reduced, creating a collection of points closer to the current point. All the mesh size

parameters (initial size, contraction and expansion factor) are defined on the toolbox GUI, before starting the optimization.

See below an example for a 2 variables problem, which forms a pattern of 4 points (note that in this example, the mesh size is 1).

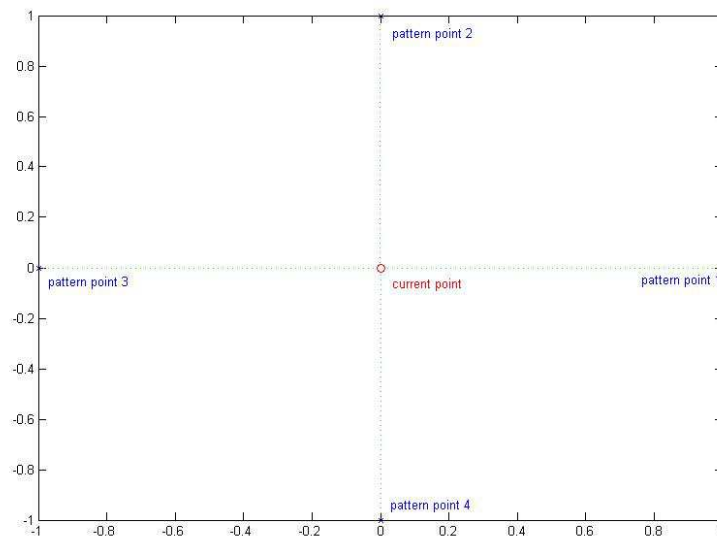


Figure 2.14: example of a set of points forming a pattern

As the mesh gets smaller and smaller when the search is close to the optimum point and the algorithm cannot find a better value of the objective function, the stopping criteria, also defined from the GUI, will determine when the optimization task stops. The minimum mesh size is set at 10^{-3} , smaller mesh sizes mean almost insignificant changes. Some useful plots, such as the current best function value, current mesh size, current best point, can be enabled on the GUI before starting the solver.

Once the optimization is launched, the main MATLAB function that calculates the lift (the objective function), including all the dos commands which

run the ANSYS Workbench in *batch* mode and the Python Scripts that control the workflow, will start, so the ANSYS will run in every function count at every iteration, will write the value of the parameters required, until any of the stopping criteria is reached, what will conclude the optimization (see the following flowchart).

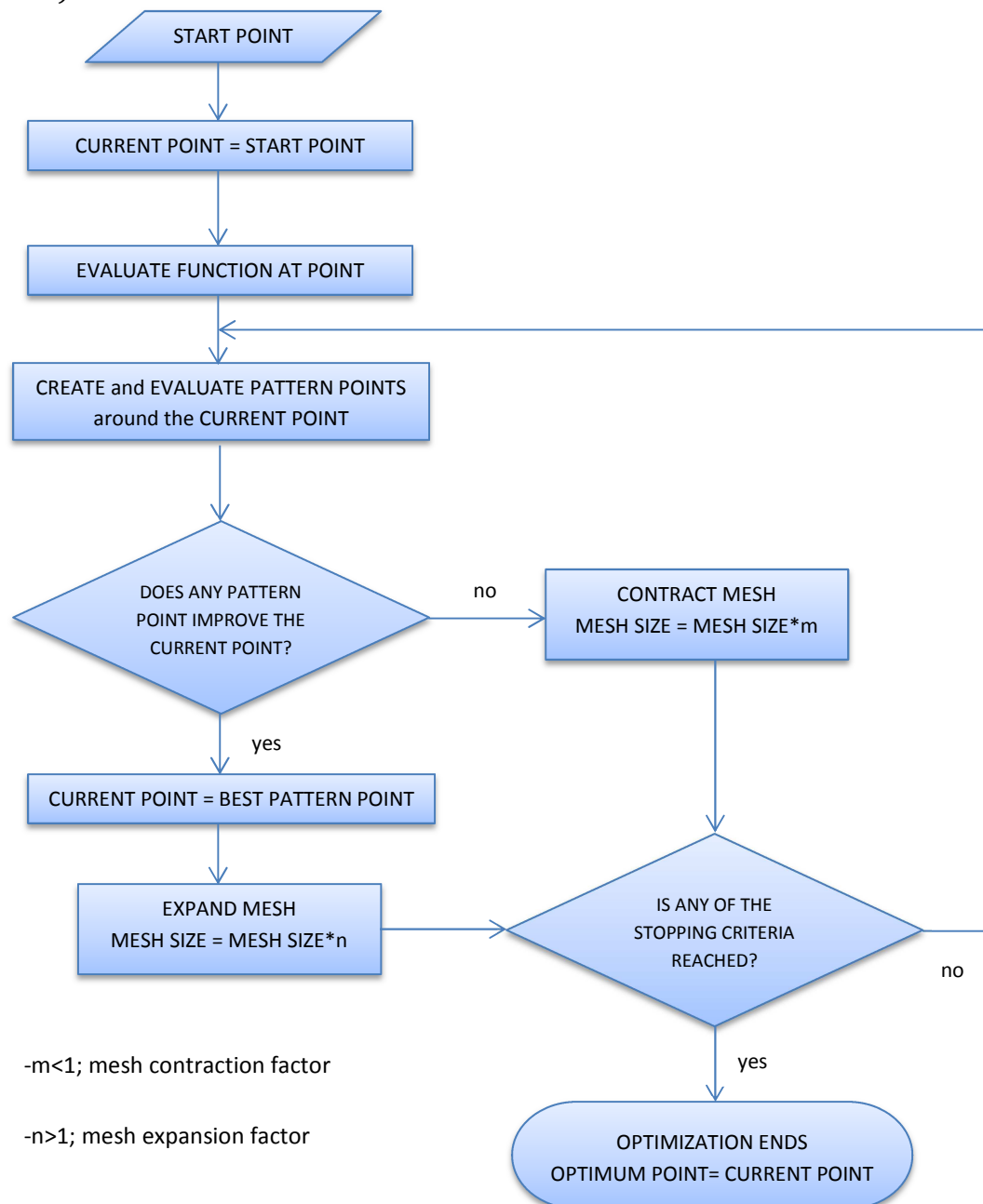


Figure 2.15: flowchart for the *Pattern Search* algorithm

3

RESULTS

AND

DISCUSSION

3. RESULTS AND DISCUSSION

After the ANSYS solver and all the options from the GUI have been chosen, the optimization can be started. The start point firstly taken is the one that define the initial profile (again see figure 2.2). The default contraction and expansion factor are kept (0.5 and 2 respectively). The mesh size tolerance is set to 10^{-3} .

This is the plot that shows the airfoil geometry evolution during the optimization.

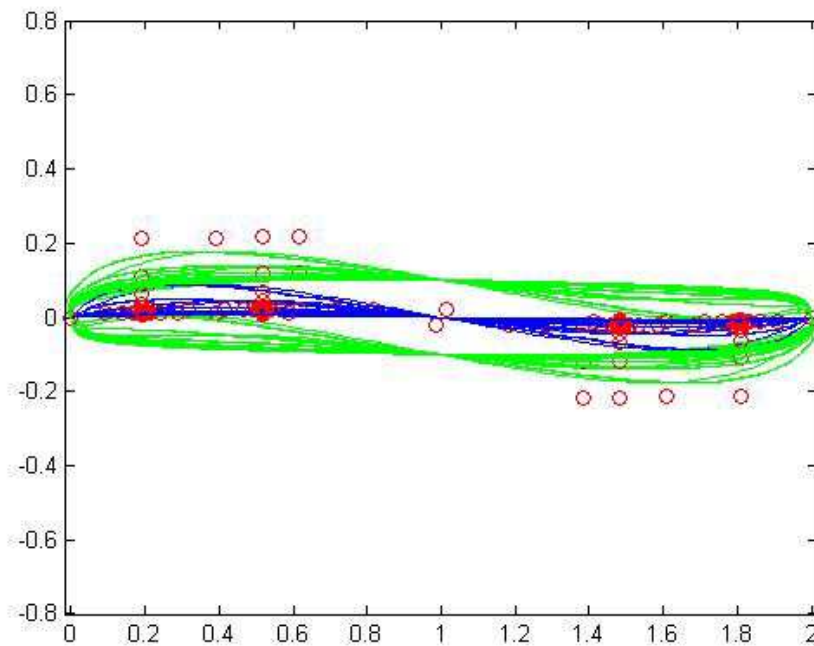


Figure 3.1: Airfoil evolution during the optimization

It can be observed that a very few S-shaped profiles were evaluated, since their values for the lift are always smaller than in the case of flat airfoils. Iteration by iteration, the solver changed the profile to a flattened design.

In the following image, the evolution of some interesting parameters of the optimization is shown:

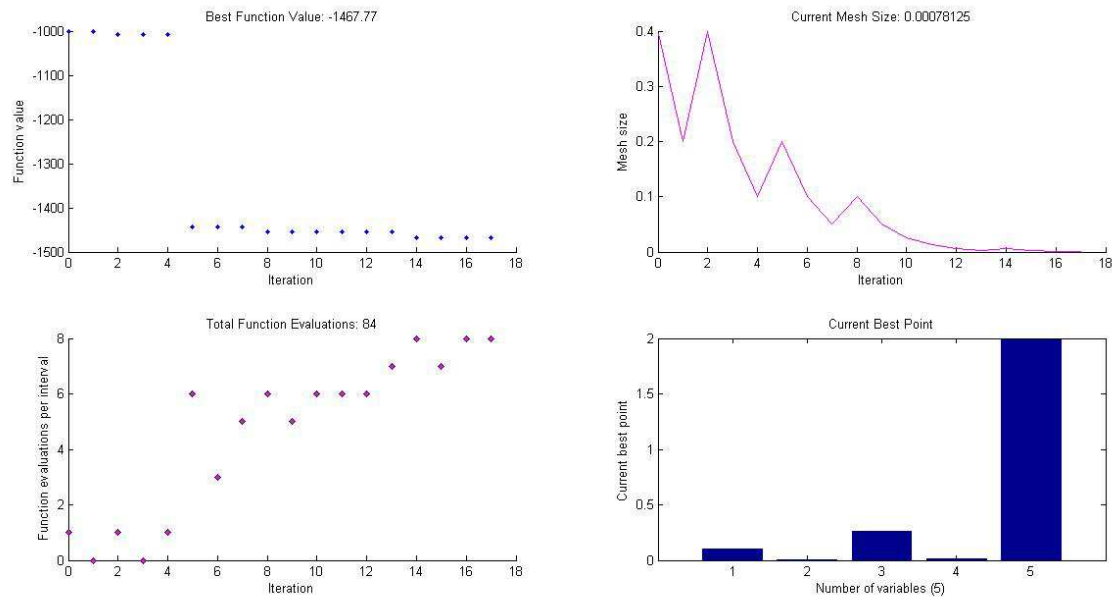


Figure 3.2: progress of the optimization parameters

The first plot (up left) shows the evolution of the objective function value, the value to be maximized (in this case, minimized, note the minus sign). The second plot (up right) is the progress of the mesh size, which shows the ability of the optimization to find better points at each iteration. An increase of the mesh size means a better point has been found and vice versa (as it was previously developed in the section 2.5). The third plot (down left) shows the number of function evaluations done at each iteration. One iteration means that all the pattern points around the current point in this iteration have been evaluated. The first iterations had a smaller number of evaluations due to a big mesh size, and the existence of boundaries left some of the pattern points out. Finally, in the last plot (down right) the current best point can be seen. These bars mean the value of the

five variables of the problem during the optimization, considered in the order they were defined in the MATLAB function (i.e. x_1 , y_1 , x_2 , y_2 , L). Since this is the plot after the optimization ended, this current point is the final best point.

An isolate plot of the optimum geometry found during the process can be seen below.

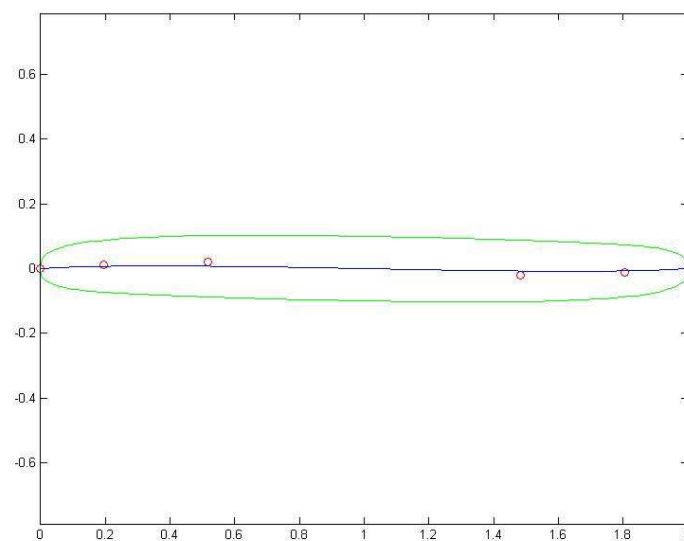


Figure 3.3: optimum profile after the optimization

The values of the variables and the results for lift, drag and lift to drag ratio are all gathered in the next table:

x_1	y_1	x_2	y_2	<i>Length</i> (m)	<i>Lift</i> (N)	<i>Drag</i> (N)	<i>lift/drag</i>
0.098	0.006	0.258	0.0100	2	1467.77	320.88	4.574

Table 3.1: variables and results for the optimum point

The complete tables with the points' coordinates, length, values of drag, lift and lift to drag ratio at every iteration are attached at the appendix.

The following plot represents the y^+ values on the upper and lower surface of the airfoil, so that the reliability of the mesh for this model can be checked again.

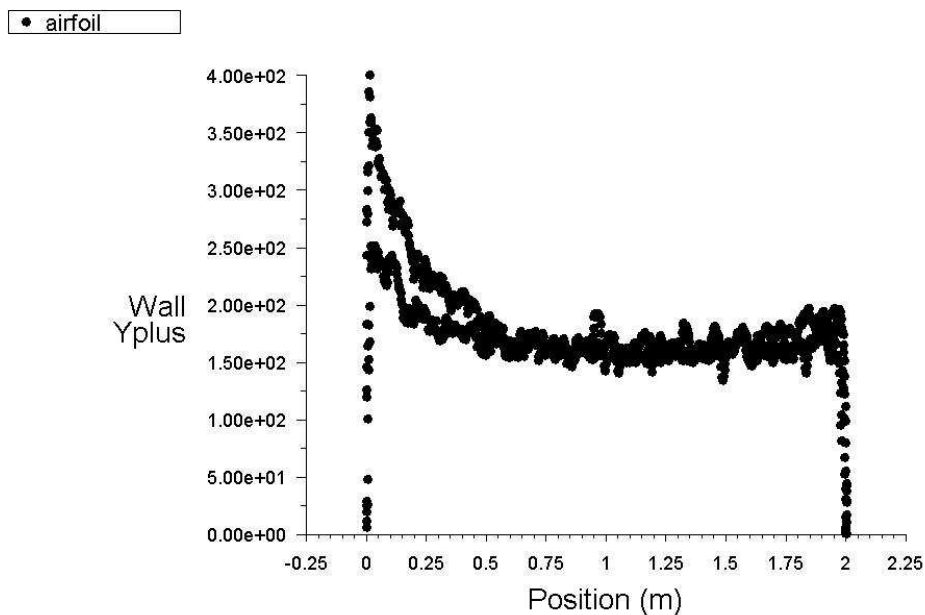


Figure 3.4: y^+ plot for the optimum design

Thus, it can be proved that the values of y^+ are placed in the correct range for standard wall functions.

A plot showing the Pressure Coefficient distribution on the airfoil can be seen below.

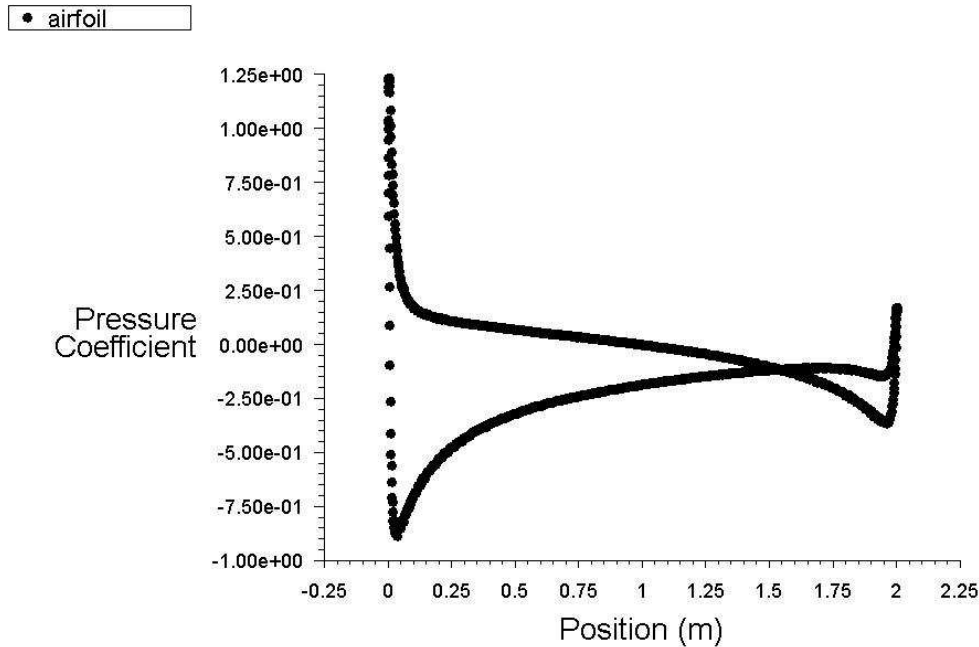
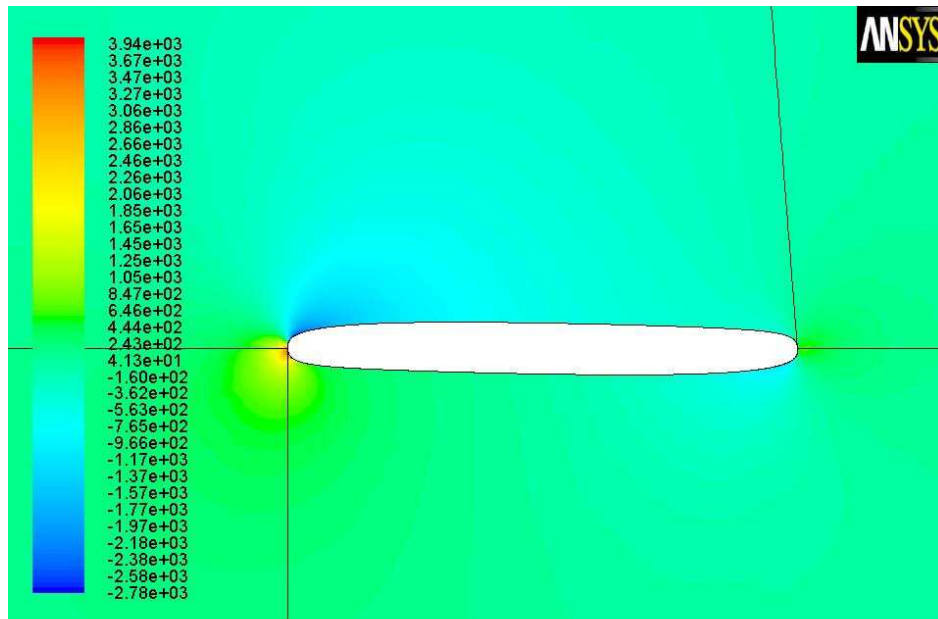


Figure 3.5: XY plot for the pressure coefficient on the airfoil

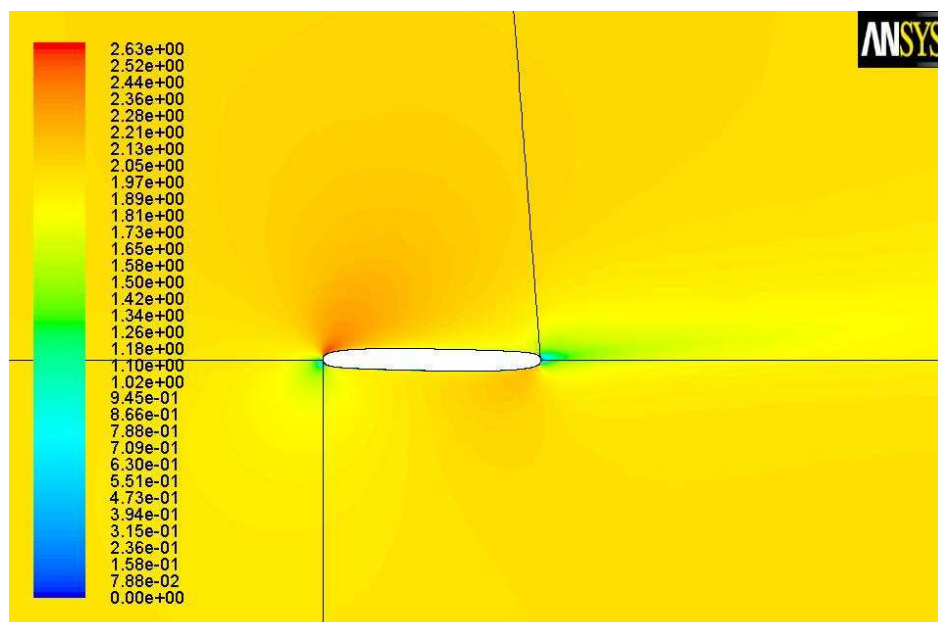
Since the upper and lower surfaces were not separately defined, the plot shows the distribution on both sides. The upper line (pressure side) corresponds to the lower surface, whereas the lower line (suction side) shows the c_p distribution on the upper surface. Computing the surface integral of the pressure coefficient on the airfoil, its average value is -0.122. Therefore, despite the maximum value of overpressure at the leading edge is greater than the maximum suction, the whole contribution of the suction in the process is bigger.

According to the last pressure coefficient plot, a figure with contours of static pressure can be seen below.



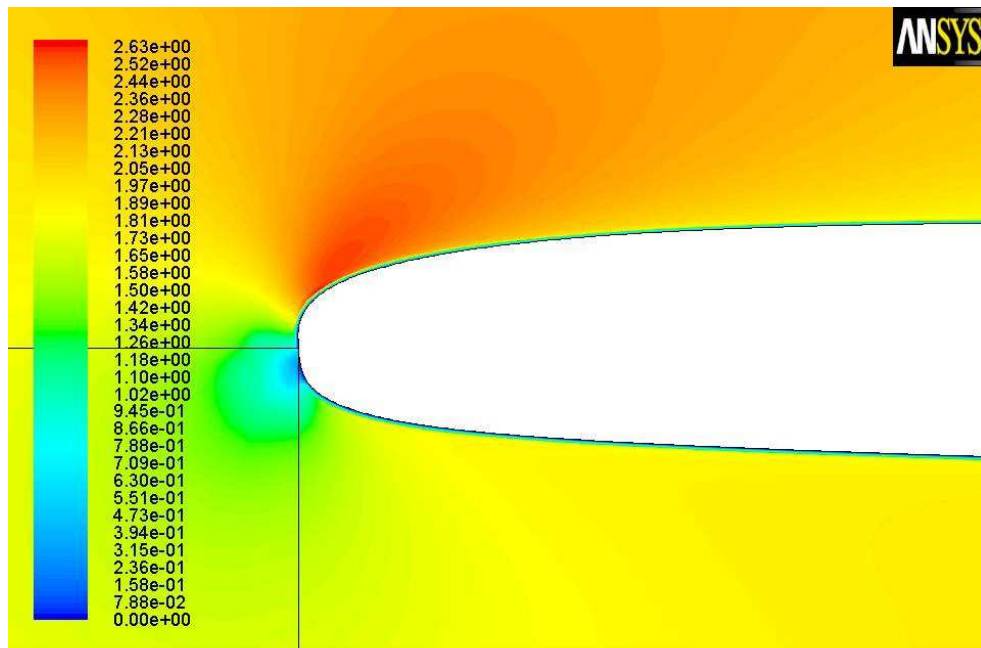
Contours of Static Pressure (pascal)

Figure 3.6: contours of Static Pressure for the optimum design



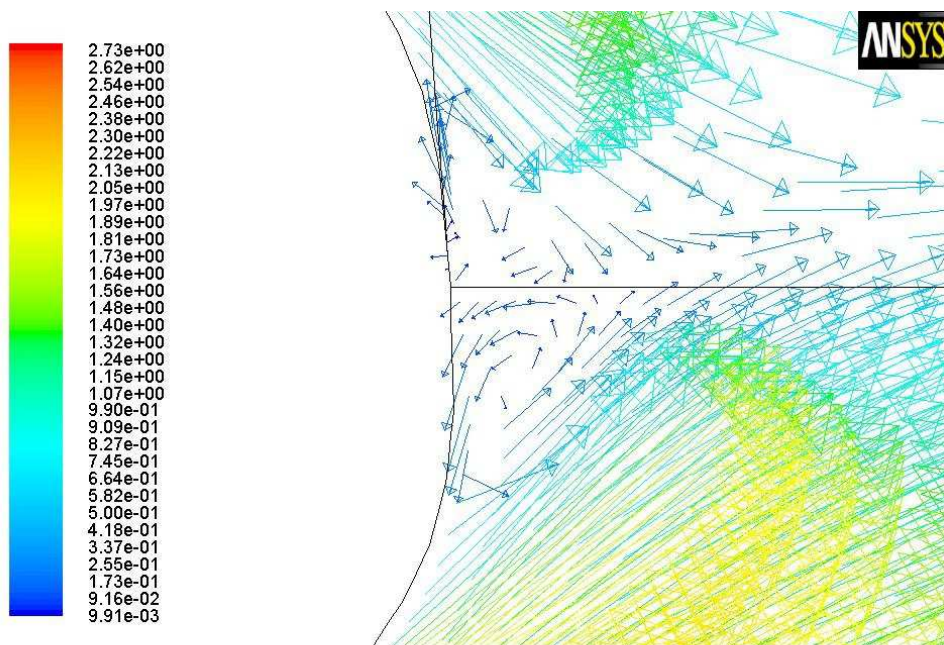
Contours of Velocity Magnitude (m/s)

Figure 3.7: contours of velocity magnitude for the optimal airfoil shape



Contours of Velocity Magnitude (m/s)

Figure 3.8: zoomed in contours of velocity magnitude at the trailing edge



Velocity Vectors Colored By Velocity Magnitude (m/s)

Figure 3.9: velocity vectors at the trailing edge

In the figure 3.7, due to the small angle of attack 5° and the flat geometry, it can be observed that the trail down the airfoil is thin and its effect disappears in a short distance. In the figure 3.8, also in consequence of the small angle of attack, the overpressure zone where the stream slows down sharply near the leading edge is closer to the horizontal axis.

Note that, despite the small angle of attack, due to the symmetrical design that leads to a rounded trailing edge (according to the leading edge), close to it appears a small double recirculation zone (see figure 3.9).

Finally, a last reflection about why the S-shaped profile (figure 2.3) did not turn out to be an optimum design.

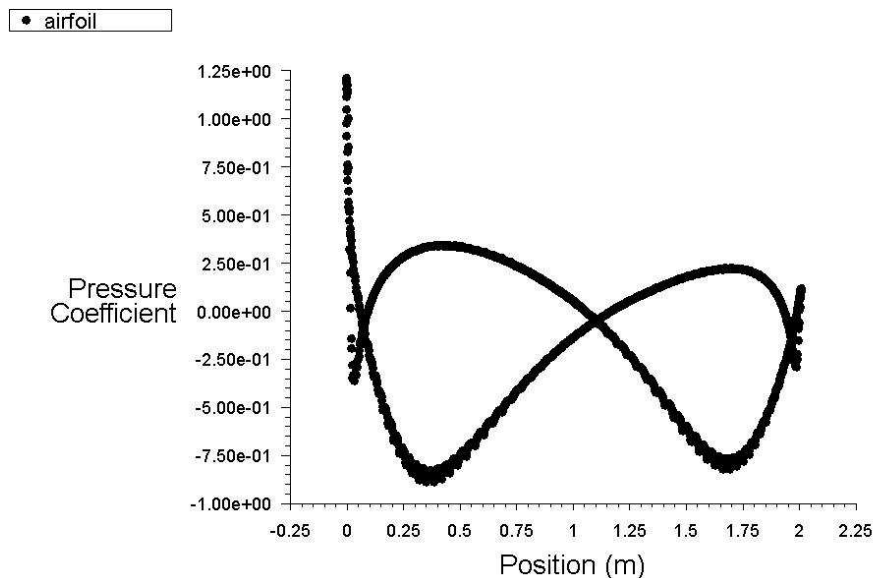
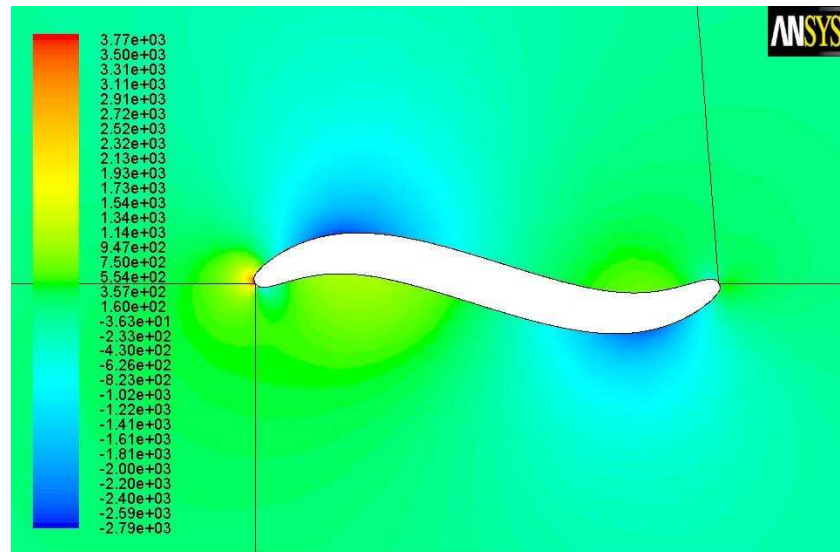


Figure 3.10: c_p distribution on an S-shape profile



Contours of Static Pressure (pascal)

Figure 3.11: contours of static pressure on an S-shape profile

With this geometry, the suction effects on both sides of the airfoil counteract each other, as well the overpressure (see figures 3.10 and 3.11). The total effect due to the suction gain from the first half gets reduced with the effect from the suction at the second half of the airfoil. In addition to this, the sharper the geometry is, the bigger the effects of drag on the airfoil are, and together with the effects on the value of the lift, lead to a considerably smaller lift to drag ratio, which is the most accurate parameter to determine whether a design has an optimum performance or not.

4

CONCLUSION

AND

FURTHER

RESEARCHES

4. CONCLUSION AND FURTHER RESEARCHES

The objective of all this work has been the development of an automatic search for the optimum design of an airfoil shape, which is used in underwater applications.

This shape is controlled by the coordinates of two points, which are the variables of all the process. Once the functions and scripts are done, linking all the workflow (C++, MATLAB and ANSYS), it lets the user get the values of lift and drag by inserting the value of these input variables from the MATLAB command window.

The final aim is being able to launch an optimization task from MATLAB's optimization toolbox, which throughout certain solvers and algorithms, manages to automatically find the optimal design, given some criteria. The goal is maximizing the lift, controlling the value of the lift-to-drag ratio as well.

The initial concept was an S-shaped design for the profile. Nevertheless, as the optimization progressed, it could be seen how the optimum design tended to a flattened shape. In fact, the sharper a profile was, the poorer the performance was. It does mean that the initial design is no longer considered an optimum design given the current circumstances. The flat design is equally valid as it keeps the symmetry feature, as well the possibility to perform in the environment it was meant for.

Here are a few future considerations based on the problems during the development of this study. The main problem appeared during the task of importing the geometry to the ANSYS Design Modeler. The lack of some kind of commands and operations via scripting (such as booleans, the drawing of curves, sketching, etc.), led to a domain that had to be partially patched step by step. It was difficult to find the most possible homogeneous mesh, and some of the geometries were conflictive during the meshing. In some cases, the current project could not converge easily, in some others the meshing tool was unable to perform it, therefore the CFD tool could not even start. If on upcoming ANSYS versions some of these commands are available for the scripting, the problem then will be possibly solved. Otherwise, in order to get further with this research, another CFD environment could be used.

Some future researches are below proposed:

- Use of a multipurpose solver, which makes possible to do simultaneously the maximization of the lift and the minimization of the drag.
- Comparison of more different viscous models on ANSYS Fluent and validation with wind tunnel tests for choosing a reliable turbulence model.
- Optimization of several 2D-sections of a rotor blade consisting of symmetrical airfoils. Then Simulations with 3-D geometry. Use of a different CFD such as ANSYS CFX.

- Study of the influence of other parameters, such as the inlet velocity, water properties (density, temperature, viscosity), etc.

5

REFERENCES

5. REFERENCES

[1]<http://www.voith.com/en/products-services/hydro-power/ocean-energies/tidal-current-power-stations--591.html>

[2]<http://www.oceanenergycouncil.com>

[3]Technology White Paper On Ocean Current Energy Potential on the U.S. Outer Continental Shelf (May 2006). Minerals Management Service. Renewable Energy and Alternate Use Program. U.S. Department of the Interior

[4]Where will ocean current energy work?
<http://www.oceanenergycouncil.com/index.php/Ocean-Currents/Where-will-ocean-current-energy-work.html>

[5]"Fundamentals applicable to the utilization of marine current turbines for energy production" A.S. Bahaj and L.E. Myers. Renewable Energy 28

[6]<http://www.haeturbines.com/hydrokinetic-renewable-power.html>

[7]<http://geography.about.com/od/physicalgeography/a/oceancurrents.htm>

[8]"Reversing free flow propeller turbine". US Patent Application Publication.

Other sites of interest:

<http://ocsenergy.anl.gov/guide/current/index.cfm>

<http://www.haeturbines.com/underwater-currents.html>

<http://www.alternative-energy-news.info/technology/hydro/tidal-power/>

<http://science.howstuffworks.com/environmental/earth/oceanography/ocean-current.htm>

APPENDIX

APPENDIX

Tables of results at every function evaluation:

x_1	y_1	x_2	y_2	length	lift	drag	l/d
0.200	0.120	0.300	0.100	2.000	560.260	476.00	1.177
0.195	0.006	0.508	0.010	2.000	1000.74	266.77	3.751
0.195	0.006	0.308	0.010	2.000	1007.94	270.46	3.727
0.195	0.006	0.508	0.010	2.000	1000.74	266.77	3.751
0.295	0.006	0.308	0.010	2.000	1262.61	260.34	4.850
0.195	0.106	0.308	0.010	2.000	776.770	368.91	2.106
0.195	0.006	0.408	0.010	2.000	1291.40	279.26	4.624
0.195	0.006	0.308	0.110	2.000	1100.95	287.29	3.832
0.095	0.006	0.308	0.010	2.000	1444.22	320.54	4.506
0.195	0.006	0.208	0.010	2.000	1307.39	281.93	4.637
0.295	0.006	0.308	0.010	2.000	1262.61	260.34	4.850
0.095	0.006	0.508	0.010	2.000	967.030	298.25	3.242
0.095	0.006	0.108	0.010	2.000	1107.81	318.03	3.483
0.195	0.006	0.308	0.010	2.000	1007.94	270.46	3.727
0.095	0.106	0.308	0.010	2.000	1040.89	406.15	2.563
0.095	0.006	0.408	0.010	2.000	1348.96	312.99	4.310
0.095	0.006	0.308	0.110	2.000	918.740	315.89	2.908
0.095	0.006	0.208	0.010	2.000	1227.01	315.49	3.889
0.145	0.006	0.308	0.010	2.000	1227.01	315.49	3.889
0.095	0.056	0.308	0.010	2.000	1248.11	343.29	3.636
0.095	0.006	0.358	0.010	2.000	1347.59	315.55	4.271
0.095	0.006	0.308	0.060	2.000	1245.18	313.67	3.970
0.045	0.006	0.308	0.010	2.000	1217.86	343.95	3.541
0.095	0.006	0.258	0.010	2.000	1453.84	324.52	4.480
0.195	0.006	0.258	0.010	2.000	1338.41	282.55	4.737
0.095	0.106	0.258	0.010	2.000	1047.86	411.51	2.546
0.095	0.006	0.358	0.010	2.000	1347.59	315.55	4.271
0.095	0.006	0.258	0.110	2.000	895.630	318.80	2.809
0.095	0.006	0.158	0.010	2.000	1218.60	317.60	3.837
0.145	0.006	0.258	0.010	2.000	1218.60	317.60	3.837
0.095	0.056	0.258	0.010	2.000	1315.46	348.35	3.776
0.095	0.006	0.308	0.010	2.000	1444.22	320.54	4.506
0.095	0.006	0.258	0.060	2.000	1169.18	312.86	3.737
0.045	0.006	0.258	0.010	2.000	1273.25	348.36	3.655
0.095	0.006	0.208	0.010	2.000	1227.01	315.49	3.889
0.120	0.006	0.258	0.010	2.000	1227.01	315.49	3.889
0.095	0.031	0.258	0.010	2.000	1165.03	321.32	3.626
0.095	0.006	0.283	0.010	2.000	1406.42	312.04	4.507
0.095	0.006	0.258	0.035	2.000	1228.92	313.50	3.920
0.070	0.006	0.258	0.010	2.000	1188.27	327.05	3.633
0.095	0.006	0.233	0.010	2.000	1416.19	321.61	4.404
0.108	0.006	0.258	0.010	2.000	1299.28	308.86	4.207

0.095	0.019	0.258	0.010	2.000	1131.04	316.70	3.571
0.095	0.006	0.271	0.010	2.000	1132.26	313.05	3.617
0.095	0.006	0.258	0.022	2.000	975.010	307.47	3.171
0.083	0.006	0.258	0.010	2.000	1133.93	318.44	3.561
0.095	0.006	0.246	0.010	2.000	1403.16	321.82	4.360
0.101	0.006	0.258	0.010	2.000	1391.73	317.26	4.387
0.095	0.012	0.258	0.010	2.000	1186.37	314.48	3.772
0.095	0.006	0.264	0.010	2.000	969.560	306.32	3.165
0.095	0.006	0.258	0.016	2.000	921.860	306.92	3.004
0.089	0.006	0.258	0.010	2.000	1150.00	315.10	3.650
0.095	0.006	0.252	0.010	2.000	1269.29	316.43	4.011
0.095	0.006	0.258	0.004	2.000	1373.82	319.97	4.294
0.098	0.006	0.258	0.010	2.000	1467.77	320.88	4.574
0.095	0.009	0.258	0.010	2.000	1167.26	314.16	3.715
0.095	0.006	0.261	0.010	2.000	1167.26	314.16	3.715
0.095	0.006	0.258	0.013	2.000	942.330	305.68	3.083
0.092	0.006	0.258	0.010	2.000	1460.34	323.36	4.516
0.095	0.003	0.258	0.010	2.000	1460.34	323.36	4.516
0.095	0.006	0.255	0.010	2.000	1460.34	323.36	4.516
0.095	0.006	0.258	0.007	2.000	1188.64	312.07	3.809
0.104	0.006	0.258	0.010	2.000	1264.16	309.92	4.079
0.098	0.012	0.258	0.010	2.000	1067.40	309.11	3.453
0.098	0.006	0.264	0.010	2.000	1171.71	308.81	3.794
0.098	0.006	0.258	0.016	2.000	1067.27	308.36	3.461
0.092	0.006	0.258	0.010	2.000	1460.34	323.36	4.516
0.098	0.006	0.252	0.010	2.000	1325.68	316.13	4.193
0.098	0.006	0.258	0.004	2.000	659.550	297.03	2.220
0.101	0.006	0.258	0.010	2.000	1391.73	317.26	4.387
0.098	0.009	0.258	0.010	2.000	1145.66	311.29	3.680
0.098	0.006	0.261	0.010	2.000	1321.07	316.44	4.175
0.098	0.006	0.258	0.013	2.000	692.530	295.84	2.341
0.095	0.006	0.258	0.010	2.000	1453.84	324.52	4.480
0.098	0.003	0.258	0.010	2.000	1453.84	324.52	4.480
0.098	0.006	0.255	0.010	2.000	979.720	306.67	3.195
0.098	0.006	0.258	0.007	2.000	1038.41	308.82	3.362
0.100	0.006	0.258	0.010	2.000	1374.99	318.09	4.323
0.098	0.008	0.258	0.010	2.000	1277.41	315.24	4.052
0.098	0.006	0.260	0.010	2.000	921.260	304.70	3.024
0.098	0.006	0.258	0.012	2.000	822.460	300.63	2.736
0.097	0.006	0.258	0.010	2.000	1430.82	319.46	4.479
0.098	0.004	0.258	0.010	2.000	1430.82	319.46	4.479
0.098	0.006	0.256	0.010	2.000	824.350	300.14	2.747
0.098	0.006	0.258	0.008	2.000	1331.01	314.62	4.231
0.098	0.006	0.258	0.010	2.000	1467.77	320.88	4.574