ISTANBUL TEKNİK ÜNİVERSİTESİ

FLUENT CAPABILITIES IN SOLVING THE FLOW PAST AN AIRFOIL

Viscous study of the NACA 4412 airfoil

STAGE DE FIN D'ETUDES DE L'ESTACA

BUCKINGHAM Sophia

12th of December 2007

AKNOWLEDGEMENTS

I would like to thank and express my sincere appreciation to my supervisor, Prof. A. Yükselen for his help, supervision and useful suggestions throughout my internship at ITU. His support and continuous guidance enabled me to complete my work successfully. I am also thankful to Resul Açıkyol for helping me during the numerical part of this study.

Finally, I express my gratitude and deepest love to my family for their endless support and everything they have done for me.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

The aim of this project was to determine what the capabilities of *Fluent* are. This software deals with Computational Fluid Dynamics and we have simulated flows past an airfoil under inviscid and viscous conditions.

First of all, the numerical results for the inviscid case were compared to those provided by the theoretical Karman-Trefftz transformation. A program based on this theory was written and allowed us to obtain the properties of the flow. In addition, it calculated the dimensions that the CFD domain should have in order to limit the error introduced by these limited boundaries. As a result, the impact of the domain on the final values was limited to an acceptable error. Thus, by respecting this criterion, the numerical and theoretical results' comparative study was made more accurate. Indeed, any deviation could then only be attributed to the software's limitations. We were able to focus our attention on the influence of *Fluent's* parameters to determine what would be the best set up in this situation. As an outcome, by applying the best suited parameters we obtained very close results and the Karman-Trefftz transformation was proved to be a useful comparative tool.

Secondly, the viscous simulations gave us the evolution of the lift coefficient with respect to the flow's incidence. In this case the flow becomes highly turbulent at high angles of attack and the flow separates. The numerical results' accuracy depends on the ability of the assigned turbulent model to reproduce these turbulent effects and therefore to measure the correct amount of separation. We decided to solve the flow surrounding the NACA 4412 airfoil and to compare the results to the experimental values obtained by R. M. Pinkerton. Based on the lift curve, we could determine which of the turbulence models is best suited for this type of application. Indeed, the models were as reliable as long as the flow remained attached which corresponds to small incidences. However, at higher angles the kω-SST turbulence model was able to correctly evaluate the separation region. To conclude, we made a comparative study between the pressure distributions provided by the kω-SST model and the tests. The distributions appeared to be very similar for most of the incidence range.

ABSTRACT

This study's primary objective was to determine what are the capabilities and limitations encountered by the CFD software *Fluent* while solving the flow past an airfoil. To do so, the project is composed of two separate parts. Indeed, we wish to investigate Fluent's potential under inviscid and viscous conditions so that the range of possible flows is covered. For both cases, we need a valuable reference from which the results are considered as being very reliable. The objective is to determine the best suited *Fluent* simulation parameters to use in order to optimize the numerical results. As a reference for the inviscid part, we wrote a program that calculates the flow properties around an airfoil, based on the theoretical Karman-Trefftz transformation. We may also recuperate the appropriate size of the CFD domain to avoid that these boundaries have a significant effect on the CFD results. The pressure distributions along the airfoil's surface were compared and the results appeared to be practically identical. As for the viscous calculations, the lift coefficients of two different turbulence models were compared to past experimental results obtained while conducting wind tunnel tests for the NACA 4412 airfoil. The aim was to determine which of the turbulence models was the most capable of predicting the correct amount of separation. The kω-SST model was very efficient and the pressure coefficients along the surface were quite similar to the experimental distributions.

Key words: capabilities, limitations, airfoil, inviscid, viscous, turbulence, pressure, distribution,

RESUME

Le principal objectif de cette étude est de déterminer quels sont les capacités et limitations du logiciel de CFD Fluent, lors de la résolution de l'écoulement autour d'un profil d'aile. Pour cette évaluation, nous avons divisé l'étude en deux parties indépendante. En effet, nous souhaitons connaitre les capitées de Fluent a résoudre un écoulement visqueux mais également non visqueux. Pour cela, nous avons besoin d'une référence fournissant des résultats précis auquel nous pouvons comparer les résultats numériques. Le but est de déterminer quel est la meilleure configuration des paramètres de simulation qui nous permet de nous rapprocher au mieux de ces résultats de référence. Pour les calcules non visqueux, nous avons écris un programme qui repose sur la transformation de Karman-Trefftz et qui nous fournis les propriétés de l'écoulement. Il est également possible d'obtenir les dimensions optimisées du domaine CFD. Ainsi, l'erreur relative introduite par ces frontières devient négligeable. La comparaison des répartitions de pressions le long du profil est très satisfaisante puisque les courbes sont pratiquement identiques. En ce qui concerne les calculs visqueux, nous avons compare deux modèles de turbulence aux résultats expérimentaux obtenu par R. M. Pinkerton lors d'essais en soufflerie. Le modèle kω-SST fut le plus performant puisque les prédictions de séparation se sont avérées précises.

Mots clés: capacités, limitations, profil, visqueux, Karman-Trefftz, erreur, répartition, pression

INTRODUCTION

The main objective of this study is to thoroughly investigate the capabilities of the CFD software *Fluent* and outline the limitations it encounters for an airfoil in particular situations. We wish to make this study as broad as possible and therefore we will make both inviscid and viscous numerical calculations. Indeed, in reality viscous forces play a major role on the flow properties especially close to a body's surface. They do not only affect the body's close surroundings but also what is further away from it. Phenomena such as separation regions, vortices, turbulent boundary layers…etc, are all caused, in their own way, by the presence of viscosity and affect the performance of an airfoil. Although an inviscid calculation is unrealistic, it can be used in many ways. The major interest offered by the inviscid assumption is that the calculations are considerably simplified. The modified equations manipulated by Fluent are much more straight forward and faster to solve. Inviscid results may be used in many ways. For example, the influence of the boundary layer on the external flow can be taken into account by considering the potential flow past the displacement body, which is the original body plus a predetermined solid layer of thickness. Both situations are equivalent but performing the inviscid calculation allows us to save important computational effort.

In the first part of the project we focused our attention on the inviscid capabilities of the CFD solver. Yet, we still need reliable results to refer ourselves to and compare the numerical results with. The best possible reference for this case can only be based on theoretical grounds. Such a method has been developed in the past and is known as the Karman-Trefftz transformation. The airfoil shape is initially defined by a few characteristics that are used during the transformation process. For us to be able to use this theory we will write a program that applies the transformation and provides the user with the flow properties. We will also use the Karman Trefftz transformation to determine the size that the CFD domain should have to insure that the pressure coefficient contains a relative error predefined as acceptable. Consequently, we will then be able to investigate the effect of *Fluent*'s inviscid parameters on the results. The objective is to get as close as possible to the Karman Trefftz results, considered as the exact values of the inviscid flow surrounding an airfoil.

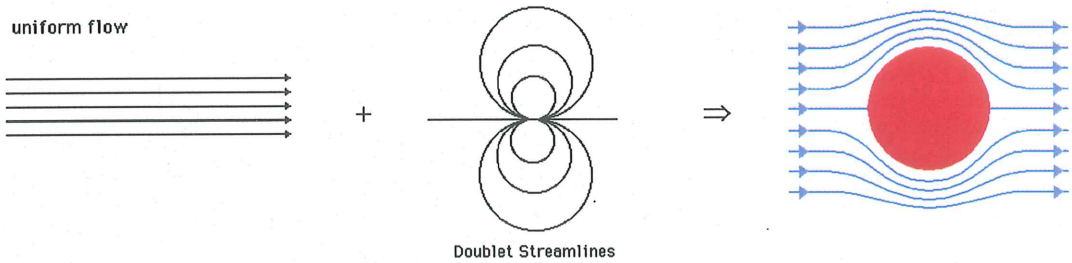
In the second part, the viscous effects are not neglected and the real flow is solved. This time there is no need for us to refer to theory since experimental results are the closest and most accurate values we may found to the flight conditions. The flow surrounding the NACA 4412 airfoil is solved, over a range of angle of attacks. Unlike the inviscid simulations, the flow is much more complex since it will appear to be highly turbulent. Therefore, *Fluent*'s performance now depends almost entirely on its capacity to reproduce these turbulent effects. The key parameter to the simulations' success is the turbulent model, thus we will compare the performance of the kω-SST and kε-Realizable turbulence models. Our aim is to determine which is best suited to airfoil applications. The pressure coefficient distributions along the airfoil's surface are critical values to airfoil design. Consequently, they are chosen as the main comparative data between the models and the experimental results.

I. KARMAN-TREFFTZ AIRFOILS

1) Presentation

a) Exact solutions for potential flows around airfoils

The Karman-Trefftz airfoils are very useful since they provide an exact solution via a complex transformation of the potential flow around an airfoil. The first successful airfoil theory was developed by Mr. Joukowski. We will exploit the theory of complex variables by representing a two dimension potential flow using an analytical function of a complex variable. It has been shown that an airfoil is obtained by applying a conformal transformation to a circular cylinder. The mapping of the complex potential flow around the cylinder will then, after transformation from the ζ complex plane to the real z plane, induce a corresponding flow around an airfoil. A cylinder is analytically composed of several simple cases. Indeed, it is an association of a uniform flow at incidence α and of a doublet of strength μ. A doublet is a useful flow obtained by letting the distance between the two sources of a source-sink pair tend to zero. The streamlines of a doublet flow are a family of circles all tangent to the origin. Therefore, a circular cylinder results from these two as follow:



The resulting flow is non-circulatory and irrotational with no vorticity, so the local angular rate of rotation given to the flow is zero. Consequently, to provide a theoretical model for the flow around an airfoil, we are missing the generation of lift since 'spinning' is needed to generate the pressure difference between the top and bottom surfaces of a wing. It is introduced by superposing a to the doublet that is placed in a uniform flow. The lift will be directly proportional to the circulation but its value can be assigned arbitrarily, leading to an infinite number of solutions. This leads us to consider the fundamental Kutta condition.

b) Kutta condition

Up to now, we have not fixed the value of circulation; therefore, there will be an infinite number of lifting flows. The Kutta condition will choose a specific flow which is the closest to the 'real' viscous flow. In order to give a physical explanation of this phenomenon, we need to introduce the role played by the viscosity of a real fluid. Without any circulation, the two stagnation points $(V = 0)$ are located on the upper surface and lower surface.

This results in a physical incomprehension if we consider the path followed by a particle. It starts by travelling on the lower surface of the profile, and then makes a U turn at the trailing edge to join the upper side. The fluid must accelerate around the edge which actually requires infinite velocity which is very unlikely in a viscous fluid. It eventually leaves the profile at the second separation point; *figure 1* illustrates what is occurring. This is when viscosity intervenes; it will damp the sharp velocity

gradient created at the trailing edge and cause the flow to separate. A flow behaving this way is totally unrealistic. The introduction of circulation in the theory is therefore fundamental. At a particular value, the rear stagnation point is positioned exactly at the trailing edge. It is only under this condition that the flow will leave the upper and lower surfaces smoothly as shown on *figure 2*. When satisfied, the Kutta condition ensures that the vorticity generated at the trailing edge is zero. Thus, we will be able to determine the lift distribution around the airfoil.

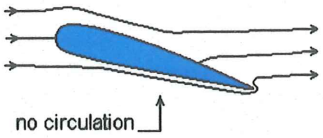


Figure 1: No circulation imposed

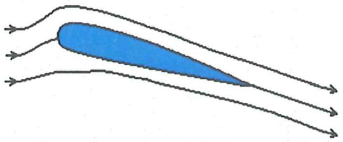


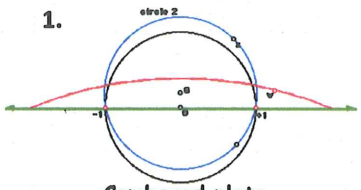
Figure 2: Kutta condition satisfied

c) Karman-Trefftz transformation

Up to now, we have established that a cylinder to which we add circulation will, after mapping, result in the desired airfoil. Let us now have a closer look at the mapping process.

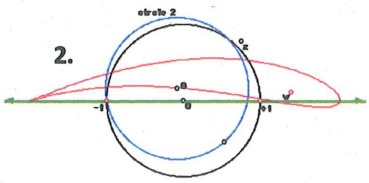
In Joukowski's method, the transformation is expressed as follow: $\xi = z + b^2/z$. This operation transforms a centred circle of radius *'a'* into a flat plate of length *'4b'*. The resulting shape is therefore 1D for the moment. In order to get closer to the desired airfoil shape, we first seek to obtain a 2D shape after transformation. To obtain an ellipse, we have to impose $a > b$ in order to generate thickness. However, the ellipse does not have the main geometrical characteristic of an airfoil, clearly distinct leading and trailing edges. In addition, the possibility of introducing camber is also fundamental so that non symmetrical airfoils may be created. For the moment the profile is incapable of generating any lift when the flow has no incidence since the lower and upper surfaces are identical. The most fundamental aspect of the transformation is that the intersections of the cylinder with the x axis will always correspond to the leading and trailing edges of the airfoil. The following geometrical explanations are based on this fact.

By shifting the cylinder upwards, camber is created. Indeed, according to the principle we just stated, the portion of cylinder above the x axis will correspond to the upper surface of the airfoil after mapping. As for the lower surface of the profile, it corresponds to the cylinder part under the x axis. This operation has resulted in a longer and a smaller portion. Consequently, the translation upwards creates an asymmetry known as camber.



Cambered plate

We observe that when we applied $a > b$, the increase in *'a'* does not only allow us to control the thickness but also the shape of the airfoil edges. Initially, for the flow plate case, the circle intersected twice with the x axis at ±b, creating infinitely sharp edges. The outcome is identical after **1.**, the camber operation. When the radius becomes larger than b, and contains ±b, roundness is introduced. Based on these two observations it becomes obvious that for the leading edge the cylinder needs to include $+b$ and that for the trailing edge it will go through $-b$.



Leading and trailing edges

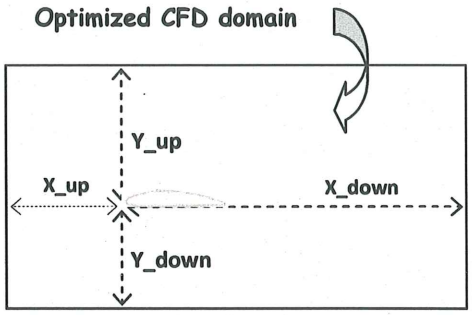
As mentioned earlier on, the Joukowski transformation was the first theoretical solution introduced. Unfortunately, it suffered from a major drawback. It does not apply to airfoils of arbitrary shapes. Indeed, the geometrical profile obtained by the mapping process provides us with a trailing edge at a zero including angle. It is known as a cusped trailing edge. This configuration is unrealistic since in reality all the airfoils present a finite angle. Indeed, such airfoils would be impossible to manufacture. The Karman-Trefftz method includes an additional parameter τ which represents the trailing edge included angle. The transformation is modified and takes into account the new airfoil characteristic, its expression is the following:

$$\xi = 2m\frac{(z+b)^m + (z-b)^m}{(z+b)^m - (z-b)^m} \qquad with: \quad m = 2 - \tau/\pi$$

2) Computational domain

a) Prediction of the domain size

When CFD calculations are undertook, the dimensions of the CFD domain are of major importance. We will be imposing boundary conditions on those edges, corresponding to the free-stream conditions. Consequently, if they are not carefully defined so that the distances between the airfoil and the boundaries are incorrect, it will affect the results irreversibly. The flow field's outer boundary, on which the uniform flow conditions are given, needs to be at a minimum distance from the airfoil. If the domain is not sufficiently far from the geometry, the results obtained from the simulation contain errors.



However, this problem can be overcome with the help of analytical airfoils. The Karman-Trefftz and Joukowsky airfoils that we just introduced give us a way of relating the lift coefficient of the airfoil to the size of the CFD domain. The exact nature of this relation will become clear once the practical procedure begins, that is thanks to the programming that is presented later on. However, we know that the free-stream conditions are theoretically at an infinite distance from the airfoil. Evidently, while the geometrical characteristics of the simulation are to be defined, the outer boundaries have to be fixed at a finite distance from the profile. We could be positioning them very far from it in order to approach real conditions but at a great cost. The computational effort would be unnecessarily high since time and memory would be wasted. A distance too large is not desired but on the other hand if it is too small the calculated coefficients will contain errors. This leads us to choose a reasonable error of free stream velocity for which the coefficients' error is acceptable and results in an optimized CFD domain as represented on the schema. Let us now determine the impact of the free-stream velocity error on the lift and pressure coefficients, C_P and C_L respectively:

$$C_P = \frac{p - p_\infty}{1/2\rho_\infty V_\infty^2} \quad \& \quad C_L = \frac{L}{1/2\rho_\infty V_\infty^2} \qquad where: \begin{cases} p: static\ pressure \\ p_\infty: freestream\ pressure \\ \rho_\infty: freestream\ density \\ V_\infty: freestream\ velocity \\ L: lift\ force \end{cases}$$

The first step is to fix the accepted relative error of free-stream velocity $\pm\Delta V_\infty / V_\infty$, resulting in the minimum distances (X_{up}, X_{down}, Y_{up}, Y_{down}). As a consequence, the error on the lift coefficient becomes:

$$C_L \pm \Delta C_L = \frac{2L}{\rho_\infty (V_\infty \pm \Delta V_\infty)^2} \simeq \frac{2L}{\rho_\infty V_\infty^{\,2}} \left(1 \pm \frac{\Delta V_\infty}{V_\infty}\right)^{-2} \simeq C_L \left(1 \pm 2\frac{\Delta V_\infty}{V_\infty} \pm \ldots\right)$$

By doing an approximation of first degree we obtain: $\left|\frac{\Delta C_L}{C_L}\right| \cong 2\left|\frac{\Delta V_\infty}{V_\infty}\right|$

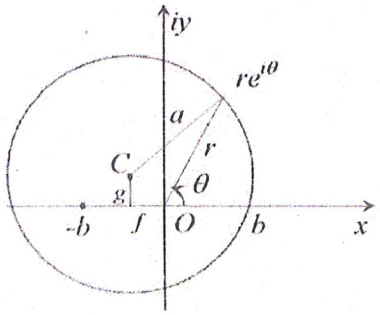
As for the error on the pressure coefficient we apply the same method and approximations:

$$C_P \pm \Delta C_P = \frac{2(p - p_\infty)}{\rho_\infty (V_\infty \pm \Delta V_\infty)^2} \simeq \frac{2(p - p_\infty)}{\rho_\infty V_\infty^{\,2}} \left(1 \pm \frac{\Delta V_\infty}{V_\infty}\right)^{-2} = C_P \left(1 \pm 2\frac{\Delta V_\infty}{V_\infty} \pm \ldots\right) \quad \Rightarrow \quad \left|\frac{\Delta C_P}{C_P}\right| \cong 2\left|\frac{\Delta V_\infty}{V_\infty}\right|$$

The results are the same so if the relative error of the free stream velocity is equal to 0.001 then the relative error of the pressure and lift coefficients is 0.002. As we will see in the more detailed description supplied by the presentation of the *Matlab* programs, we have based our calculations on the accepted value of the pressure coefficient's relative error. It has been fixed to: $\left|\frac{\Delta C_P}{C_P}\right| = 0.005$

b) Karman-Trefftz calculations

Karman-Trefftz airfoils are designated by six numbers. By groups of two numbers, they refer to the camber ratio, the thickness ratio and the maximum camber point in this very order. For instance, if the studied airfoil is called KT 001540, it corresponds to γ = 0% (camber ratio), τ = 15% (thickness ratio) and a maximum camber point located at x_{max}/c =40% of the airfoil's chord. However, all the formulas we will be manipulating for the determination of the airfoil coordinates, velocities... refer to three geometrical parameters rather than γ, τ and x_{max}/c. As we can see on the schema representing the cylinder they are equal to distances (f, g, b). ,



Therefore, we need to find a way of relating the two descriptions, so that we can name the airfoil correctly but also conduct all the necessary calculations by using the geometrical parameters which are **F**, **G** and **m**. F and G are non-dimensional parameters related to the thickness and the camber of the airfoil respectively, and defined as: $G = g/b \quad \& \quad F = f/b$

The thickness ratio of a Karman-Trefftz airfoil depends mostly on parameter F, and the camber ratio depends on G. Parameter m affects the trailing edge and also the position of the maximum camber point. An iterative method has been developed by *M.A. Yukselen* and *M.Z. Erim ('A general iterative method to design Karman-Trefftz and Joukowski airfoils', see references)*. It allows us to obtain the values of F, G and m for a Karman-Trefftz airfoil for a certain thickness, camber ratio and trailing edge angle (or maximum thickness point position). Tabulated results are available and have been used for our calculations.

We will now give a more detailed description of the mathematical expressions we had to manipulate to obtain the pressure distribution of the domain. Indeed, the analytical Karman-Trefftz transformation had two major interests.

First of all, the determination of the CFD domain dimensions which will be useful for all the inviscid numerical simulations. To do so, we need to be able to calculate the pressure coefficient C_P at any point within the domain. As we introduced previously, the boundaries are fixed so that C_{P_MAX} along each of the four lines composing the rectangular domain has a tolerated relative error approximately equal to 0.005.

Secondly, we are particularly interested in the pressure distribution along the airfoil's surface. The aim is to compare the analytical distribution to the numerical solution. The Karman Trefftz transformation deals with a potential flow so an inviscid model is applied in *Fluent* to solve the flow. In both parts, the objective is identical and that is calculating the pressure coefficients.

- **Karman-Trefftz transformation → coordinates**

To determine the airfoil coordinates, the calculation is straight forward. First the cylinder is defined, and if we refer to the previous schema, the geometrical relation giving the distance *'r'* can be found. The points are defined in a complex plane so that we deduce the coordinate of each point along the translated cylinder. Finally, the airfoil coordinates are directly deduced by applying the Karman-Trefftz transformation from the z plane to the ξ plane:

$$R = G\sin\theta - F\cos\theta + \sqrt{1+2F+(G\sin\theta - F\cos\theta)^2} \qquad with: R = {}^{r}/_{b}$$

$$z = b.(R\cos\theta + i.R\sin\theta)$$

$$\xi = mb.\frac{(z+b)^m + (z-b)^m}{(z+b)^m - (z-b)^m}$$

As for the calculation that results in the CFD domain, lines along which C_P will need to be calculated are defined in the airfoil ξ plane. However, for the following steps leading to the pressure values, we need to have the corresponding coordinates in the z plane.

An inverse Karman-Trefftz transformation is applied to the series of points $\xi(x_\xi, y_\xi)$ defining the straight lines:

$$\boxed{z = \frac{(\xi+mb)^{1/m} + (\xi-mb)^{1/m}}{(\xi+mb)^{1/m} - (\xi-mb)^{1/m}}}$$

- **Determination of the velocity field**

To calculate the velocity field surrounding the airfoil, we will first need to calculate the velocities in the z plane V_z, determined by using the angle θ_1 shown on the scheme.

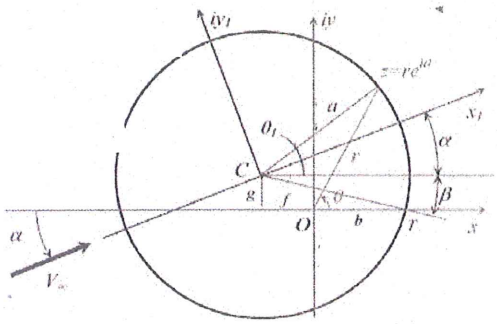


Figure 3: Cylinder used for the Karman-Trefftz transformation

We obtain the velocities along the airfoil and along the domain lines, by dividing the velocity V_z by the derivate of the coordinates ξ by z.

$$\theta_1 = \theta - \alpha_{radians} - a\sin\left(\sqrt{\left(F^2+G^2\right)/\left((1+F)^2+G^2\right)}\times\sin\left(\theta + a\tan\left(G/F\right)\right)\right) \quad \rightarrow \quad V_z = 2\left(\sin\theta_1 + \sin\left(\alpha_{rad}+\beta\right)\right)$$

$$\frac{d\xi}{dz} = 4m^2b^2\frac{\left(z^2-b^2\right)^{m-1}}{\left[(z+b)^m-(z-b)^m\right]^2} \quad \Rightarrow \quad V_\xi = \frac{V_z}{\left|d\xi/dz\right|}$$

- **Pressure coefficients**

The calculation is straight forward since we now have all the necessary variables to deduce the pressure coefficient at any point within the CFD domain: $Cp = 1-\left(V_\xi^2/V_\infty\right)$

c) Matlab programs

All the *Matlab* scripts can be found in *Appendix A_1/*

- **Main function: *'maintest.m'***

As we just explained, we will be creating several *m-files*, each one of them serving a particular purpose. One way to run the program is in a chronological order where the first function calls the second and so on... This formulation is not very flexible and requires that each function contains the correct syntax to run all the necessary *m-files*. In a situation where some general parameters are needed all along, it lengthens the program since they will have to be defined for each function. Instead of using this unpractical method, the program is executed from the main program which performs the compilation. It is a script as well and contains a series of commands, which results in running all the functions in the right order. This way, it is much easier to share information in between the functions. All the variables which are calculated in one of the compiled functions and are needed later are defined as an output of the script so that they become global variables. Consequently, those parameters are stored by the main program. Whenever they are needed by another function, they are available from the main program and the function just needs to define them as inputs. The previous

problem we mentioned is solved since the general parameters can be defined only once, in the main program, and positioned as inputs whenever they are needed by a function.

Let us now describe the contents of the different scripts starting by the main program. The compiled functions within the main program are only briefly introduced here. The purpose is to state the major steps and specify the input & outputs.

DATA INPUTS: the general parameters which define the Karman-Trefftz airfoil are contained in a *'.txt'* file, so the first command executed in the main program results in reading the file and acquire the information. However in this first part of the script, a choice is made on whether the incidence or the lift is imposed. One of the two parts, 1) or 2), is activated and the other commented.

1) Deducing the lift from the incidence

- The correct data input is read accordingly. The main parameters recuperated by the program are the following:

$$\begin{cases} Karman-Trefftz\ parameters: F, G, m \\ Number\ of\ points: N \\ Incidence\ in\ degrees: AlfaD \end{cases}$$

- *'Ktreff.m'* is compiled. The Karman-Trefftz transformation is applied, and transforms the cylinder into the desired Karman-Trefftz airfoil.

2) Deducing the incidence from the lift

- The lift coefficient replaces the incidence so that we have as inputs:

$$\begin{cases} Karman-Trefftz\ parameters: F, G, m \\ Number\ of\ points: N \\ Lif\ tcoefficient: Cl \end{cases}$$

- *'incidence.m'* calculates the incidence of the airfoil, necessary for the Karman-Trefftz method.
- *'Ktreff.m'* is executed as in the previous case and the transformation is applied.

3) Common part: CFD domain dimensions

- *'coeff.m'* is compiled and all the aerodynamic coefficients are calculated.
- *'domain.m'* is compiled and will determine the numerical boundaries.

4) Figures

Figure 1: Representation of the cylinder in the z plane. The circle has been shifted upwards and left according to the analytical airfoil parameters *(F, G, m)*.

Figure 2: Karman-Trefftz airfoil in the ζ plane.

Figure 3: Pressure distribution along the airfoil surface.

Figure 4: Velocity distribution along the airfoil surface.

5) *Text file output*

- *'results.txt'* contains the airfoil coordinates, lift and pitching moment coefficient.

6) GAMBIT journal

- In the following part of our study, we will compare the *matlab* results of the analytical method to numerical inviscid calculations. For the airfoil geometry we create in *Gambit* to be sufficiently accurate, a large number of points are required. This program is used, and in particular the *'ktreff.m'* function which calculates the airfoil coordinates. In order to obtain a smoothly shaped profile we will define 500 points. To facilitate the task, we avoid having to enter all the vertices manually by creating a gambit '.jou' file. By running the journal in *Gambit*, all the points are created automatically. To facilitate the procedure, an output text file is written in this part of 'maintest.m', while respecting the syntax imposed by a Gambit journal, it contains all the points along the airfoil surface.

7) Export of the pressure coefficient distribution along the airfoil

- When the comparison between analytical and numerical will take place in the inviscid part of the report, it is the pressure coefficient along the airfoil that interests us. Consequently, the analytical data will be needed, which leads us to create a *'.dat'* file containing all the useful information. The file is named according to the angle of attack of the flow.

- **Karman-Trefftz transformation: *'ktreff.m'***

```
function[z,U_z,x1_zeta,x_zeta,y_zeta,U_zeta,zeta,beta,b,XLE,Cu,Cp] =
                    ktreff(AlfaD,F,G,m,Chord,N1)
```

INPUTS: Karman-Ttrefftz characteristic parameters (**F**, **G** and **m**), the chord (**Chord**), the number of points (**N1**) and the incidence in degrees (**AlfaD**). To summarize, all the information of the program's input file plus the chord.

First the useful variables are calculated such as the flow incidence in radians, the distance b needed to calculate the leading edge distance...etc. This script corresponds to the main core of the program. Most of it is contained in *'ktreff.m'* and the analytical method is conducted.

→ The angle θ of each point in the z plane is determined according to the total number of points.

→ The cylinder coordinates (z plane) are calculated all around the cylinder, based on θ and R.

→ Thanks to the Karman-Trefftz transformation we obtain the airfoil coordinates (ζ plane).

→ The velocities along the airfoil V_ζ are calculated, which depend on V_z, determined with the circulation.

→ By using the velocity values, we deduce the pressure coefficient all along the airfoil surface.

OUTPUTS: The cylinder and the airfoil coordinates + components (x, ζ, x_ζ, $x_{1\zeta}$, y_ζ), the velocities along the airfoil (V_ζ, V_z), the square of those airfoil velocities and the pressure coefficients (C_u and

C_p). The leading edge distance (X_{LE}) , the distance **b** and the angle **β** are also placed as outputs of the function.

- **Calculation of the incidence: *'incidence.m'***

```
function [AlfaD,Gamma] = incidence(CL,F,G,m,Chord)
```

INPUTS: All the general parameters of the KT airfoil and of the flow contained in the program input file (**F**, **G**, **m** and C_L), plus the chord (**Chord**).

The aim of this script is to calculate the angle between the profile and the flow. The information we need is contained in the given lift coefficient. Indeed, a relation links the lift coefficient to the circulation and another links the circulation to the incidence, which can then be deduced. The formulas are the following and are used directly in the program:

We know that the circulation is equal to: $\Gamma = 4\pi A b \sin(\alpha_R + \beta) \quad \Rightarrow \quad \alpha_R = \sin^{-1}\left(\frac{\Gamma}{4\pi A b}\right) - \beta$

The incidence can be calculated by replacing the circulation once expressed according to the lift coefficient: $C_L = 2\Gamma/c \quad \Rightarrow \quad \Gamma = \frac{C_L \times c}{2}$

OUTPUTS: The circulation (**Gamma**) is placed as an output of the script since it represents one of the airfoil's characteristics, as well as its angle of attack to the flow, in degrees (**AlfaD**).

- **Aerodynamic coefficients: *'coeff.m'***

```
function [Gamma,CL] = coeff(AlfaD,F,G,m,Chord,beta,b,XLE)
```

INPUTS: The incidence of the flow (**AlfaD**) is needed to calculate the coefficients, which also depend on the general characteristics of the Karman-Trefftz profile (**F**, **G**, **m**) and its chord (**Chord**). Some of the variables (**b**, X_{LE}, **β**) calculated in the function *'ktreff.m'* to carry out the transformation are also required to calculate the circulation (implicating C_L). Indeed, in *1)* the incidence was among the initial inputs and therefore *'incidence.m'* was never compiled so the circulation was not calculated. This script is part of the common part of the program so it has to function in both situations.

The primary objective of the script is to determine the lift coefficient because we will need it in the following part of the study, when the effect of camber and thickness on the numerical boundaries is to be determined. The geometrical characteristics can be plotted as functions of the lift coefficient. In any case, the incidence is known so the previous formulas appearing in the *'incidence.m'* script are used directly. First the circulation is calculated and then the lift coefficient is deduced thanks to the second relation linking both variables. The second part of the function is optional, and can be activated if the user is interested in the pitching moments. We have all the necessary information to calculate it at the leading edge and quarter chord. To access the pitching moment values they have to be added to the outputs.

OUTPUTS: The circulation (**Gamma**) and the lift coefficient (C_L) are needed as outputs.

- **Computational boundaries: *'domain.m'***

```
function[IX,JY,Cp_max] = domain (m,b,AlfaD,F,G,N1,Gamma)
```

INPUTS: All the general parameters contained in the input file read at the very beginning of the main program *'maintest.m'* are needed. Therefore, all the airfoil characteristics (**F**, **G** and **m**), the angle of attack (**AlfaD**) and the number of points (**N1**) are all recuperated by defining them as inputs of the script. The distance **b** is required to perform the inverse transform and the circulation (**Gamma**) is used to calculate the velocities in the z plane.

Let us now describe how the function is able to determine the locations of the 4 numerical boundaries. First of all, each line is considered one at a time. For instance, if the front vertical line is to be determined, its length is firstly defined. We usually fixed it around 100 so that the entire region influenced by the airfoil is surely considered. Then its initial position along the X axis is entered. The rest of the script functions automatically. It is based on a loop that repeats itself as long as a certain condition is not satisfied. Within the loop, a series of equally spaced points along the line are converted to the z plane thanks to the inverse Karman-Trefftz transformation. Then the velocities V_ζ are calculated to finally deduce all the pressure coefficients along the line. We are searching for the position where the maximum C_P value does not exceed the accepted relative error of 0.005. However, this condition cannot be exactly respected so the tolerated values are contained in a small interval around 0.005 equal to $[0.0047;0.0053]$. If C_P is smaller than 0.0047, the result is too accurate and we need to be more economic. As a result the boundary is shifted slightly closer to the profile. In the opposite situation where $C_P \geq 0.0053$, the relative error is larger than accepted, so the line is shifted further away from the airfoil. These tests are conducted until the condition satisfied. The procedure is repeated for the four lines surrounding the rectangular domain. All the numerical data provided by the Matlab program based on the Karman-Trefftz transformation has been placed in *Appendix A_2/.*

OUTPUTS: In case we need to access the C_{P_MAX} along the limit, it is defined as an output. The two defining parameters of the line, its length (**JY**) and its position along the axis (**IX**) are exported to the main program.

3) Post processing of the results

We have investigated what would be the impact of the thickness and camber ratios of an airfoil on the resulting numerical domain's dimensions. In order to observe their effect independently, one of the two geometrical parameters is held constant while the other increases. Thanks to the information provided by the iterative method, we could have the geometrical characteristics for each combination of the following thickness and camber ratios: $\gamma = [0\%;5\%;10\%;15\%]$ & $\delta = [5\%;10\%;15\%;20\%]$

These airfoils to which we applied the Karman-Trefftz transformation regrouped thin to thicker shapes, from symmetrical to more cambered profiles. We began by observed the evolution of each of the four distances as functions of the flow incidence. However, a better indicator of the airfoil

behaviour is to plot those distances with respect to the lift coefficient. This significant aerodynamic characteristic can be related to the domain. It shows us to what extent the airfoil impacts on its surroundings, according to the lift it provides.

a) CFD domain as a function of the flow incidence

It appeared that the general trend of each of the four dimensions was the same. Consequently, only the results of the distance between the airfoil and the upstream boundary are shown here. All the other results can be found in *appendix B*. In order to determine the general effect of the two ratios on X_{up}, let us observe the following curves:

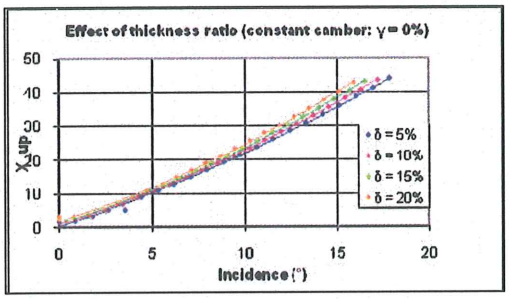


Figure 4: γ=0% and δ ε (5%,10%,15%,20%)

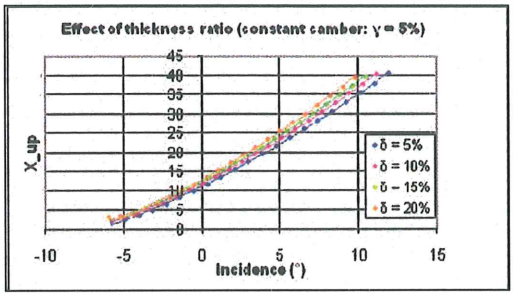


Figure 5: γ=5% and δ ε (5%,10%,15%,20%)

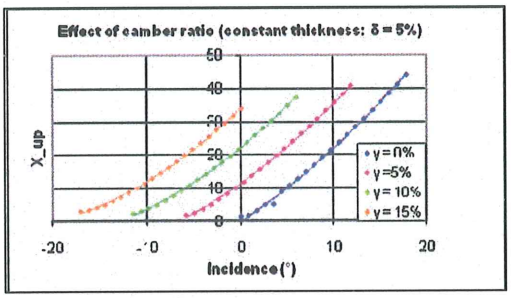


Figure 6: δ=5% and γ ε (0%,5%,10%,15%)

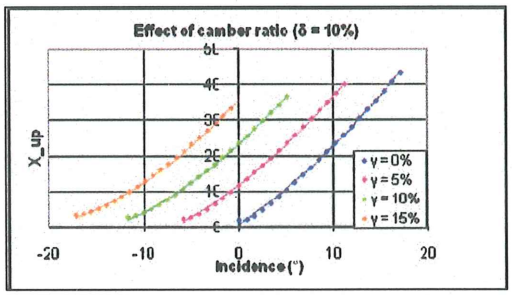


Figure 7: δ=10% and γ ε (0%,5%,10%,15%)

- Effect of thickness variation

The first two figures each report the evolution of four identically cambered airfoils; their camber ratios are respectively equal to 0% and 5%. Both figures illustrate X_{up} as a function of the incidence for four different airfoils:

$$\gamma = 0\% \rightarrow (KT\ 000540;\ KT\ 001040;\ KT\ 001540;\ KT\ 002040)$$

$$\gamma = 5\% \rightarrow (KT\ 050540;\ KT\ 051040;\ KT\ 051540;\ KT\ 052040)$$

When the camber ratio further increases, the behaviour remains the same so we limit our observations to these two schemas. As we can see on *figure 4*, when the airfoil becomes thicker, the curve's slope increases. This still stands for *figure 5*, but the camber is different and affects the results.

- Effect of camber

Figures 6 & 7 give us a slight idea about the consequences of an increasingly cambered airfoil.

Indeed, an increment in camber translates the curve to the left, leading to a larger X_{up} for the same angle of attack. The camber ratio evolves within a wider range going from a symmetrical airfoil case (0%), to a maximum value of 15%. From this information, we can deduce that the camber of the

airfoil gives us the horizontal positioning of the curve. The following plot represents the evolution of the constant coefficient '*c*' of the curves, which equations are of this form: $X_{up}(\alpha) = a \cdot \alpha^2 + b \cdot \alpha + c$

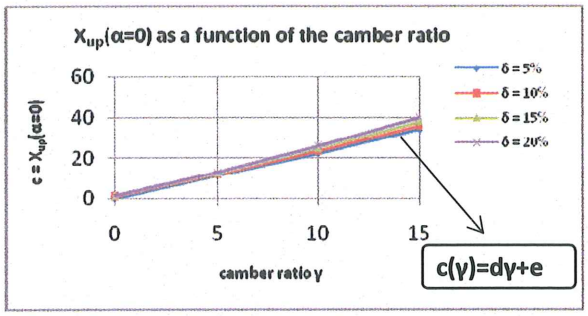


Figure 8: Evolution of the constant c = X_{up}(α=0)

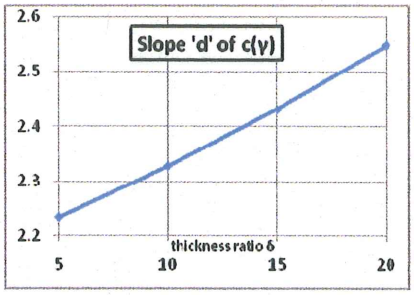


Figure 9: Evolution of the slope d = Xup(α=0)

As we can see, the intersection between the y axis and the curves, corresponding to the 'c' coefficient, mainly depends on the camber ratio. Considering these two variables as proportional is a good approximation since a linear curve fits well to the data we have collected. The thickness of the airfoil has a minor influence, which still needs to be considered if we want to be accurate in the evaluation of the appropriate X_{up} distance. Indeed, the slope of the straight lines relating γ and *c* depends on the thickness δ. The curves $c = f(\gamma)$ are approximately straight lines of equation: $\boxed{c(\gamma) = d.\gamma + e}$

The second effect of the camber variation is related to the 1st coefficient of the curves $X_{up} = f(\alpha)$. It appears that the thickness is not the only influencing parameter since the values also increases with γ.

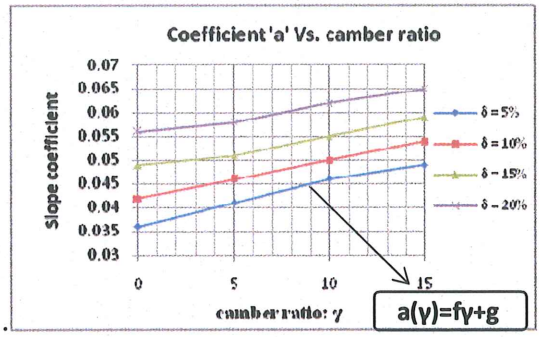


Figure 10: Evolution of 'a' as a function of γ

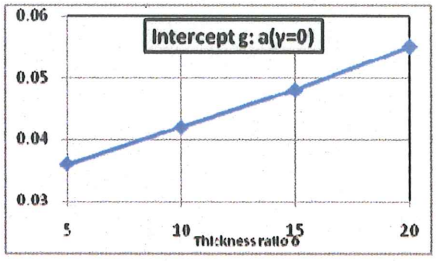


Figure 11: Evolution of intercept 'g' as a function of δ

Thus, we have plotted the evolution of 'a' as a function of the camber ratio for example. Several thickness ratios are represented on figure 10 just above. This figure illustrates the combined influences of the thickness and camber ratios. The curves of $a = f(\gamma)$ are linear fitted, so that we have a(γ) = fγ + g. The gradients (f values) are almost equal, but the value $a(\gamma = 0) = g$, increases with δ, the straight lines are being progressively shifted upwards. Therefore, on figure 11 is represented the evolution of the intercept coefficient of the straight lines $a(\gamma) = f\gamma + g$, as a function of δ.

As for the coefficient '*b*' of $X_{up}(\alpha) = a \cdot \alpha^2 + b \cdot \alpha + c$, its prediction appears to be more complicated than the two other ones. On the next page is the evolution of the coefficient with respect to the camber ratio.

To simplify the procedure, the evaluation of this coefficient is done graphically instead of analytically. For example, if δ=12% then it is contained within $[10\%;15\%]$ so we draw on the graph an average curve using the two surrounding ones. Then we deduce the coefficient 'b' directly according to the value of the camber ratio γ.

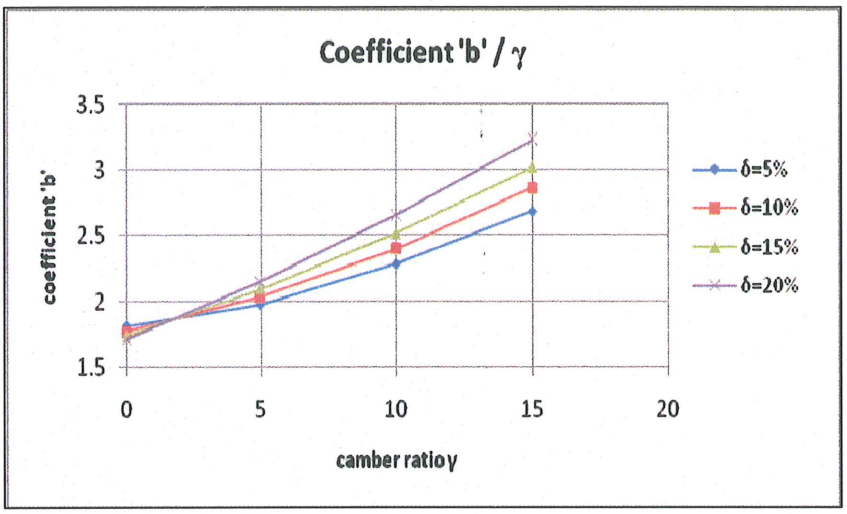


Figure 12: Evolution of 'b' as a function of γ

We have exploited all of the curves $X_{up} = f(\alpha)$, and are able to outline a method allowing use to impose the correct dimensions to the CFD domain. These distances insure a relative error on the pressure coefficient of about 0.005. First of all we summarize the different equations:

$$X_{up} = f(\alpha) \rightarrow Xup = a.\alpha^2 + b.\alpha + c \quad (b \simeq graphically)$$

$$\begin{cases} a = f(\gamma) \rightarrow a(\gamma) = f.\gamma + g \\ f \simeq 7.35\times10^{-4} \end{cases} \qquad \begin{cases} c = f(\delta) \rightarrow c(\delta) = d.\delta + e \\ e \simeq 0.57 \end{cases}$$

We can describe the different steps of the method as follow:

1. **Coefficient 'a'**: $a(\gamma) = f.\gamma + g$
   - f is approximated: $f \simeq 7.35\times10^{-4}$
   - we use the graph g(δ): $\delta \Rightarrow g$
   - from $a(\gamma) = f.\gamma + g \rightarrow \gamma \Rightarrow a$
2. **Coefficient 'b'**: b obtained graphically from the graph $b(\gamma)$
3. **Coefficient 'c'**: $c(\gamma) = d.\gamma + e$
   - e is approximated: $e \simeq 0.57$
   - we use the graph d(δ): $\delta \Rightarrow d$
   - from $c(\gamma) = d.\gamma + e \rightarrow \gamma \Rightarrow c$

Finally, $\boxed{Xup = a.\alpha^2 + b.\alpha + c \quad by\ replacing\ (a,b,c) \Rightarrow Xup/\alpha}$

These calculations were all based on the curve $X_{up}(\alpha)$ as an example. The same method applies to the three other distances, all the necessary data is in *Appendix B_1/*. Once all four calculations are finished, the numerical domain may be defined.

b) CFD domain as a function of the lift coefficient

The following curves are used if it is the relation between the CFD domain and the aerodynamic characteristics of the airfoil which is to be determined. This time, the curves will be exploited directly without any further analysis. The previous method was developed so that the numerical calculations could be done with precise values for the domain. Indeed, the software *Fluent* requires the user to specify the angle of attack while initialising the flow. At the contrary, the lift coefficient is obtained after the simulation is run. Practically speaking, it is the relation between (X_{up}, X_{down}, Y_{up}, Y_{down}) and the incidence α that is the most likely to be useful that is why we tried to develop a reliable method. Here, we are only interested in the general behaviour of the distances with respect to C_L.

The results have been plotted and the four curves can be found in *Appendix B_2/*. As an example, we rely in this paragraph on $Y_{up}(C_L)$.

- Effect of thickness

The thickness ratio of the airfoil has very little influence. To illustrate this comment, one of the boundary distances is represented as follow:

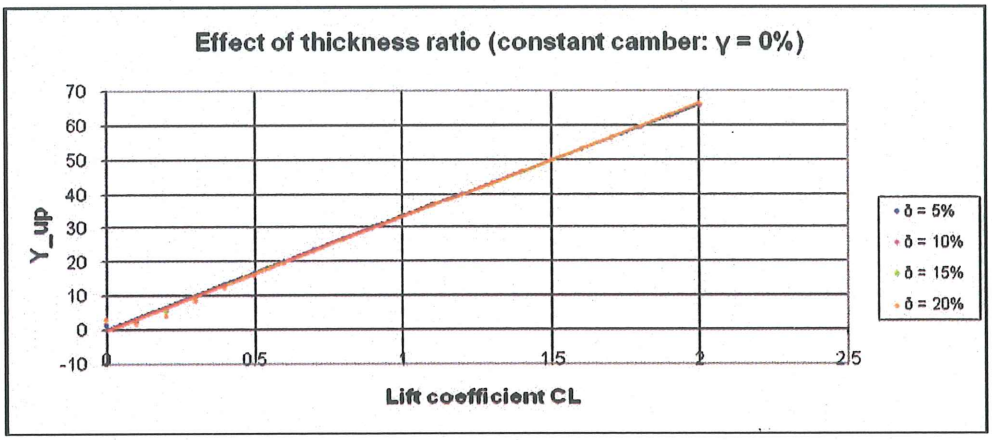


Figure 13: Effect of δ on $Y_{up}(C_L)$

The camber ratio remains constant but the thickness changes. However, as we can see the four curves are practically identical and the effect of thickness can be neglected. The same phenomenon is observed for the three other distances. $Y_{up}(C_L)$ does not depend on δ.

- Effect of camber

Unlike the thickness ratio, the camber ratio influences the behaviour of the boundaries as a function of the lift coefficient. This is only applicable for the horizontal distances since Y_{up} and Y_{down} behaviours remain constant despite the thickness variations (*ref. Appendix B_2/*). The following figure represents $X_{up}(C_L)$:

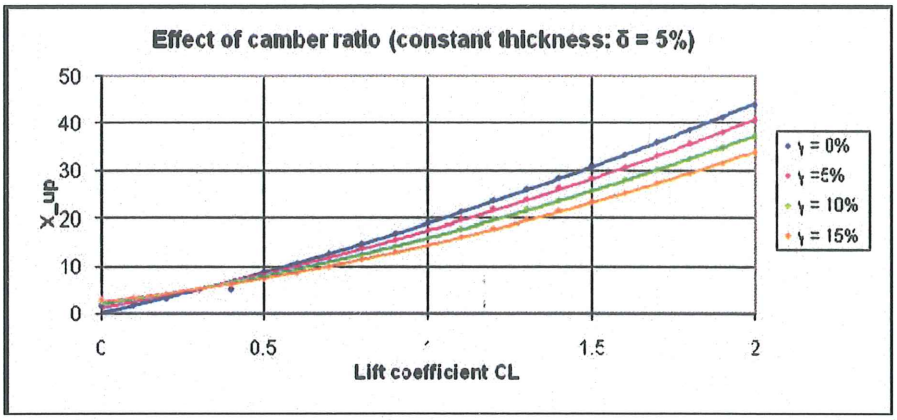


Figure 14: Effect of γ on $X_{up}(C_L)$

For a constant lift coefficient, the boundary X_{up} increases with δ. Moreover, it is the rate of increase that becomes greater. The same behaviour is observed for X_{down}..

Firstly, a greater camber will allow the airfoil to produce more lift for the same angle of attack. Secondly, an airfoil with a higher lift coefficient has as greater impact on its surroundings.

What we can deduce from these observations is that the gain in aerodynamic performance obtained by an increase in camber, has nevertheless an additional impact. Indeed, the surrounding area experiences larger pressure differences which can affect the aerodynamic behaviour of other close parts of an aircraft for instance. This side effect is exacerbated as the lift increases.

II. INVISCID CALCULATIONS

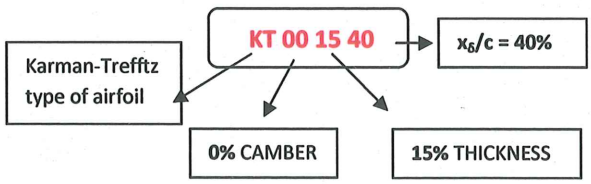
Before undertaking the numerical calculations around an airfoil, the flow past a circular cylinder was solved. All this preliminary study can be referred to in *Appendix C*. Both cases are inviscid calculations and can be closely related. Indeed, the airfoil's geometrical characteristics are not so distant from the cylinder's. They both present curved edges that lead to numerical difficulties. The problems encountered for the cylinder are very lightly to occur in our future airfoil study. Consequently, we investigated the effect of a diversity of numerical parameters that have to be specified in order to initiate the simulation. Thanks to this approach, we could improve the results considerably by solving the numerical inconsistencies. Indeed, some of the flow features we were observing were not supposed to appear in non viscous flows. In part II, the inviscid airfoil simulations largely benefitted from the previous sphere calculations. We will often refer to them and take as granted the conclusions of *Appendix C*.

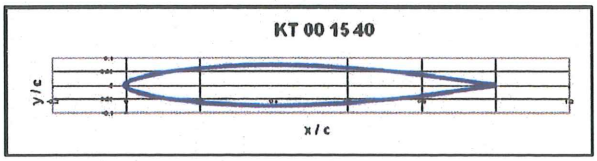
1) Mesh generation

a) Geometry

The geometry was obtained by using a 'journal file' with *Gambit,* the software we use for generating the geometry and mesh that is then exported to *Fluent*. To have a smooth enough geometry many points had to be entered manually. To avoid this lengthy task, the Journal file allows us to create the shape automatically just by important the *.txt* file to the program. All the points' coordinates are obtained with the *Matlab* program we presented in part I in order to calculate the analytical profile. This way, when the comparison between analytical and numerical data will take place, both geometries are identical. The roundness of the leading and trailing edges is ensured firstly by a large number of points (500), and secondly by the usage of the cosinus mapping function implemented in the *Matlab* program. Indeed, the analytical airfoil is obtained as we know by applying the Karman-Trefftz transformation to a cylinder, resulting in the desired airfoil. Therefore, the cylinder is created by carrying out a cosinus mapping. Despite the regularly spaced points of the circle, once projected to the x-axis, non-uniformly spaced points are generated on the airfoil. This operation is very useful since a higher concentration of points is obtained in the leading and trailing edge regions so that they are more precisely defined. We chose the airfoil **'KT 00 15 40'** as an example of the Karman-Trefftz type of airfoil. As a reminder, let us specify the significance of each number. On the left hand side are the corresponding parameters we used as inputs in *Matlab* to define the geometry:

Thickness parameter :	$F = 0.05279$
Camber parameter :	$G = 0.0$
Roundness factor :	$m = 1.87614$
Number of points :	$N = 500$





One of the critical issues to consider during the generation of the geometry was the number of points to use. At first, the airfoil was defined with 100 points, which was very insufficient. Indeed, although they are linked to each other by curves, when N is small the curves get closer to straight lines. The flow then encounters ups and downs in pressure, due to the unsmooth geometry, causing the pressure distributions to oscillate. These oscillations become increasingly important with the mesh's density since the more nodes the greater ups and downs captured by the mesh. 500 points appeared to be satisfactory; it allows the oscillatory phenomenon to disappear, and the curve's smoothness becomes grid independent.

b) Domain

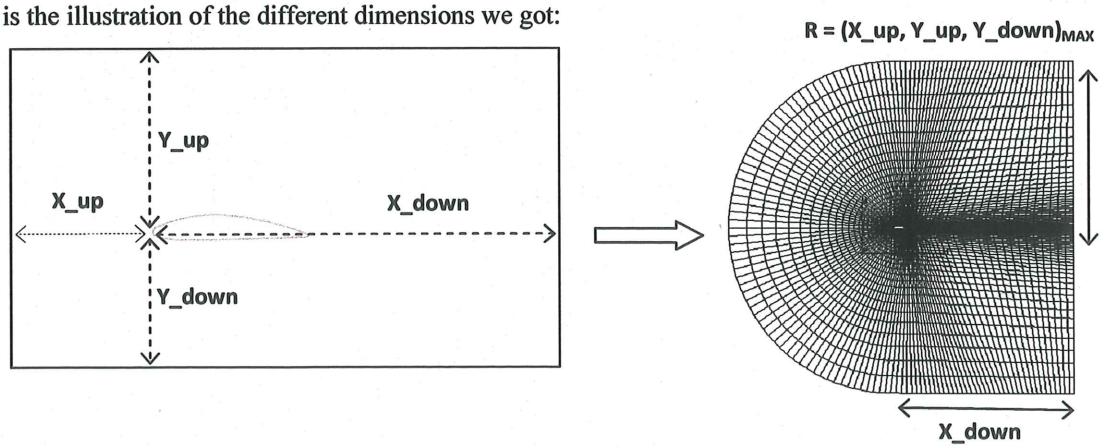
- Structured grid

For the meshing of a simple airfoil with no additional bodies (flaps...) a structured grid appears to be the best meshing technique to adopt. They present considerable advantages over other grid methods by giving to the user a high degree of control. We may decide on how the control points and edges are distributed which gives us total freedom while positioning the mesh. In addition, quadrilateral elements are very efficient at filling space and can support higher degrees of skewness and stretching before the results are affected. This allows us to create a high concentration of points within the regions of strong gradients and expand out with a less dense packing of points away from these areas. We will however need to insure good quality meshing for the cells contained in the critical areas. In addition to the advantages a structured grid offers, it typically requires the lowest amount of memory for a given mesh size due to its simple topological description. It also executes faster since structured block flow solvers are optimized for the structured layout of the grid. Finally, the post processing of the results is simplified since the logical grid planes are an excellent reference for examining the flow and plotting the results.

- C-mesh

A classical C grid is used to mesh the area surrounding the geometry and is a particularly good body-fitted grid for an airfoil. We have chosen to use a structured grid so as we were emphasizing previously, we need to focus on the meshing of the high gradients region located around the airfoil. A grid based on a C-mesh method is a good start.

Let us recall the work we previously achieved while post processing the analytical airfoil calculations. Indeed, the analytical Karman-Trefftz theory was used as a tool to optimize the future CFD study of these airfoils. While determining the optimum computational domain by calculating the pressure coefficients along each side of it, we finally obtained all the distances of the rectangular domain. This is the illustration of the different dimensions we got:



Unfortunately, the computational effort related to the domain dimensions cannot be utilised at its most since the quality of the mesh is also a fundamental aspect to consider. The use of a rectangular domain is also excluded due to the important skewness factor (evaluator of the cell's deformation) it would produce; a body-fitted grid is required. The C mesh is definitely the best solution. To conserve the original shape of this meshing technique, we will need once again to make a concession on our previous efforts to reduce the domain's size. The radial part of the mesh is defined by a unique radius, represented by the maximum value of (X_up, Y_up, Y_down). The second dimension is X_down , downstream of the flow.

c) Grid effect

- Procedure

For the grid dependency study, we will investigate the influence of the 'KT 00 15 40' airfoil on the incoming flow for an incidence of 5°. The first data we collected during the analytical work gave us the domain's dimensions as a function of both the lift coefficient (Cl) and the incidence (α). Just for this preliminary study, we decide not to simulate a flow with zero incidence to make the impact of grid variation on the solution more noticeable. The grid domain dimensions are easily calculated thanks to their equations as functions of α. We used the method developed in part I.

For this airfoil, the characteristic lengths of the computational domain are: $\alpha = 5° \Rightarrow \begin{cases} R = 21.25 \\ X\_down = 9.9 \end{cases}$

The aim is to determine the optimum mesh, for which the solution is 'grid independent'. Several grids have been generated and the inviscid simulations were run under identical conditions. The objective is to compare the pressure coefficient C_P calculated by *Fluent* in between the grids. As the mesh density increases, the values of C_P should stabilize. The objective is to decide which grid is dense enough for the solution to be sufficiently accurate. Indeed, as discussed when we undertook this procedure for the sphere, all depends on the error we are prepared to tolerate.

Now, let us first of all describe how the meshing of the airfoil was defined. The most important step of the mesh generation consists in deciding how the nodes should be distributed. The quality of the mesh entirely depends on the physical understanding of the flow. To highlight the critical areas of the mesh, let us describe one of the C_P curves we obtained.

- Typical Cp distribution

We have decided to base the grid dependency study on the distribution of pressure coefficient along the airfoil. According to the expected evolution of C_p, it allows us to compare the meshes and comment on their consistency. Finally, the aim is to reach grid independency and conclude on which mesh to use.

First of all, let us explain the shape and evolution of C_p along the two airfoil surfaces. This preliminary analyse will be useful for the choice of the final mesh. As a result of the calculation, we obtain a typical pressure distribution throughout the airfoil's surface as follow:

We often refer to the distribution of pressure over the airfoil to study its aerodynamic performance. This distribution is non-dimensionalized by the free stream dynamic pressure so that it is finally obtain the pressure coefficient: $C_p = \frac{P - P_\infty}{1/2\,\rho V_\infty^{\,2}}$. From our simulations, we obtain as expected a classical behaviour of the flow surrounding the airfoil. The positive values of C_p (upper curve)

correspond to the lower surface and vice versa. C_p starts at about 1 at the stagnation point which is located close to the leading edge on the lower surface since $\alpha = 5°$. At this point, the flow separates, part of it heads towards the upper surface while the rest goes downwards.

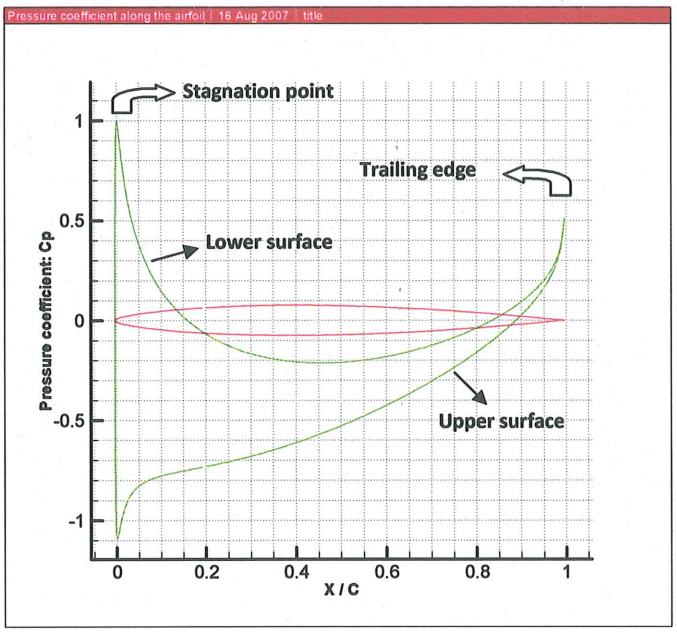


Figure 15: Typical Cp distribution

→ Upper surface: The lower curve represents the evolution of the upper surface flow, from the stagnation point on. The important drop of pressure from 1 to the lowest value of Cp corresponds to the strong acceleration experienced by the flow going over the curved nose of the airfoil. This sudden rise leads to the highest speeds towards the leading edge; consequently the flow of the upper surface then slowly decelerates. This region is known as the pressure recovery region.

→ Pressure recovery: The pressure will increase from its minimum up to the trailing edge coefficient. The flow therefore encounters adverse pressure gradients, which might cause boundary layer separation. However, we will not come across this situation since the calculation model is inviscid.

→ Lower surface: We may notice that on quite a large portion of the graph, the lower surface C_p's become negative, meaning that some suction (downwards force) is present.

Finally the flow recovers a positive value of C_p near the trailing edge where the upper and lower flows rejoin. The Kutta condition is satisfied since the rear stagnation point is located exactly at the trailing edge. This ensures that the vorticity generated at the trailing edge is zero.

Consequently, the flow around an airfoil will contain regions of strong pressure gradients. The grid we generate needs to take this into consideration; the nodes are then distributed accordingly. Only then will we be able to reach grid independency since the computational effort is limited. Indeed, if the grid does not match the flow gradients, the calculation will diverge without ever reaching a grid independent state.

- Simulation parameters

We have applied the following settings to the four different meshes:

- **Solver**: *Segregated / Implicit* → Refer to the cylinder calculations
- **Energy equation**: *Off* → $\begin{cases} V_\infty = 1\ m.s^{-1} \\ a = 344\ m.s^{-1} \end{cases} \Rightarrow M_\infty \ll 0.3$, that is low enough to assume that the flow is incompressible, consequently the energy equation is unnecessary.
- **Material**: *Air* → *incompressible* ⇒ constant density: $\rho = 1.225\ kg/m^3$
- **Operating conditions**: 101 325 Pa → Atmospheric conditions

- **Boundary conditions**:
  - Velocity inlet → $\alpha = 5° \Rightarrow \begin{cases} X = V_\infty \times \cos(\alpha) \\ Y = V_\infty \times \sin(\alpha) \end{cases} \Rightarrow \begin{cases} X = 0.9962 \\ Y = 0.8716 \end{cases}$
  - Pressure outlet → *Gauge pressure* : 0 *Pa*

- **Discretization scheme**:
  - Pressure / Velocity coupling: *'SIMPLE'*
  - Pressure interpolation: *'PRESTO!'*

Just as for the cylinder simulation preceding the airfoil study, the default interpolation scheme is not suitable for curved geometries. We are expecting a constant velocity profile at the airfoil boundary since the calculation is inviscid, however towards the leading edge the flow is slowed down at the wall, due to undesired friction effects:

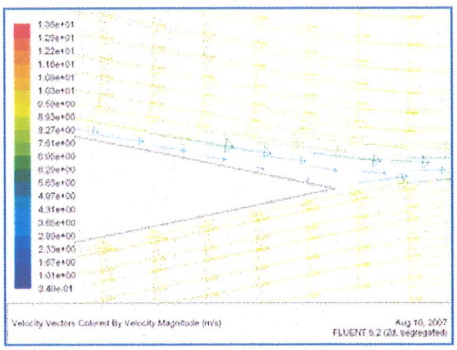


Figure 16: Standard interpolation

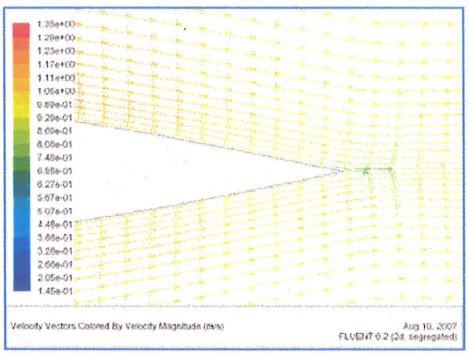


Figure 17: PRESTO! interpolation

As we can see, the PRESTO! pressure interpolation solves the problem since the velocity distribution along the surface is back to a velocity profile characterizing a potential flow, where $V_{wall} = V_{farfield}$

- Order of accuracy: *'Second order Upwind scheme'*

In order to determine whether using a second order upwind scheme is necessary or not we have compared the pressure distribution of a 1st and a 2nd order scheme. The following plot represents the evolution of the pressure coefficient along the airfoil's surface. Only the order of accuracy differs between the two curves of *figure 18*, the mesh and all the other parameters are identical. They evolve similarly along the profile but it appears that the 1st order scheme underestimates C_p all along the profile. The biggest differences are observed near the leading edge where the highest pressures are

exerted *(ref. Zoom *)*. The variation reaches 14% in that region; therefore the 1st order approximation is less accurate and we will use a 2nd order upwind scheme.

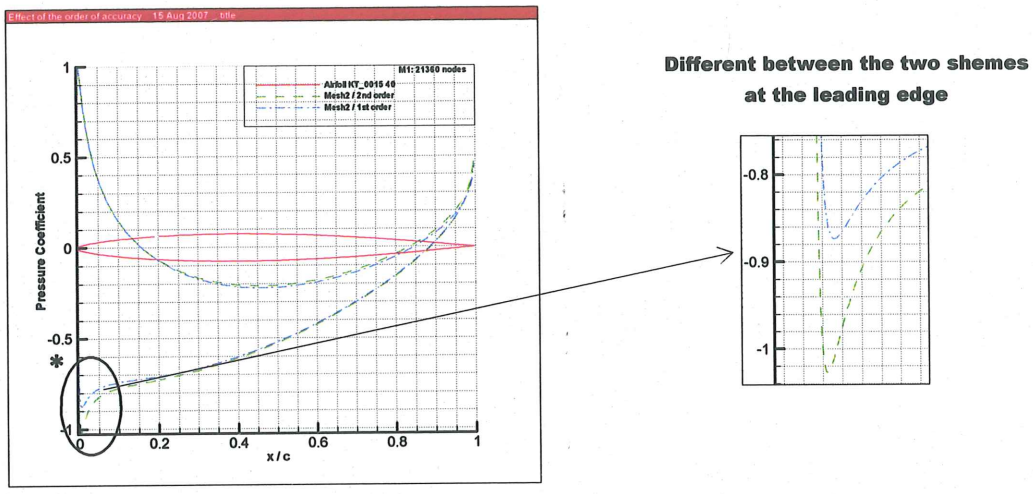


Figure 18: Comparison of the 1st and 2nd order upwind schemes

- Relaxation factor

We have opted for the segregated solver and consequently each equation which needs to be solved is associated with an under relaxation factor. This parameter is used to control the update of the computed variables, at every iteration. Its expression is the following: $\phi_{new} = \phi_{old} + \alpha \cdot (\phi_{calc} - \phi_{old})$

By reducing the under-relaxation factor, we are slowing down the rate at which the solution changes during the iteration. If we refer ourselves to the formula, the difference between the solutions at the old and new time steps is less taken into account. In this way, the fluctuations between the time steps are limited and the simulation is more lightly to converge. In this study, the simulations are not encountering any convergence difficulties.

Therefore it is unnecessary to modify the under-relaxation factors since it would only be a waste of memory. The default values are not modified and are equal to:

$$\begin{cases} Pressure: 0.3 \\ Density: 1 \\ Body\ forces: 1 \\ Momentum: 0.7 \end{cases}$$

- Residuals

At each time step, an average error is calculated for each equation to be solved.
This error is referred as the 'residual'. The smaller the value, the closer we are from obtaining a converged solution. For a 2D incompressible inviscid flow problem, there are three differential equations to be solved. This means we need to monitor three residuals for convergence: momentum, x-velocity and y-velocity.

We implement within *fluent* the residuals' order of magnitude for which the iterations are stopped and the calculation considered as converged. For all the equations, we change the default value from 10^{-3} to 10^{-5}. However the momentum equation never reaches such a level of accuracy, and the residuals stabilize at a constant value below 10^{-3} which is small enough.

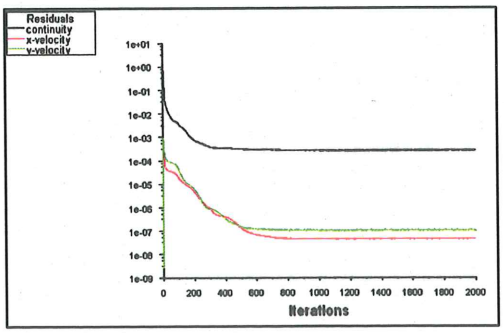


Figure 19: Scaled residuals / Converged calculation

All the simulations had similar convergence paths to the one represented on *figure 4*. At this stage, all three equations are solved and the variables have converged. Most importantly the pressure distribution along the airfoil is totally stabilized.

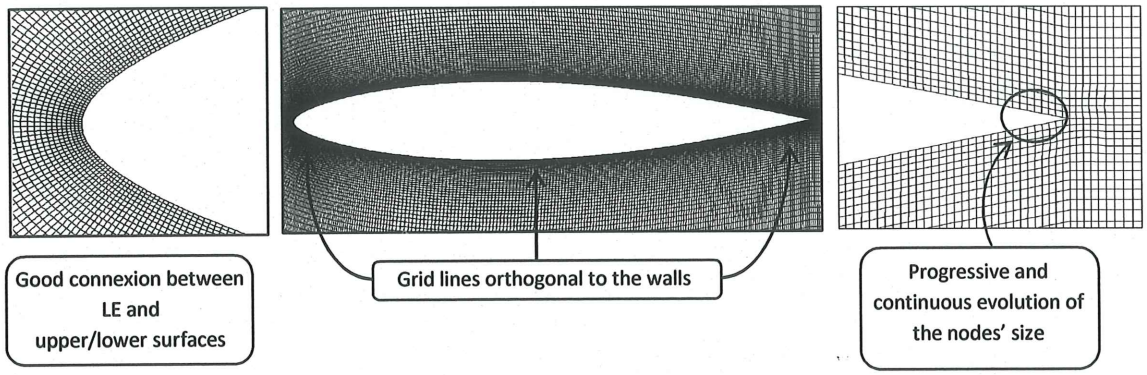
- Quality of the mesh

After this brief analyse of the pressure evolution along the profile, we can deduce which are the regions requiring particular attention. Within these areas, the node distribution is important but also the quality of the mesh. Otherwise, strong gradients and distorted cells will cause the calculation to diverge.

→ Leading edge: the stagnation point is located somewhere around the leading edge according to the flow incidence. A strong concentration of nodes is essential to collect precisely the information. To do so, we imposed a one sided length ratio from one node to the other, forcing them to become closer and closer as we approach the leading edge.

→ Trailing edge: The pressure recovery region we mentioned previously endures changes in pressure but gradually. We optimize the mesh on this portion by decreasing the number of nodes. However as we approach the rear of the airfoil, a higher concentration is required to capture the sudden changes occurring towards the trailing edge. This is also done by imposing a length ratio.

This distribution is applied symmetrically to the upper and lower surfaces; let us now emphasize the importance of a good quality meshing. Indeed, in the critical regions, it is essential that the quadrilateral cells remain as square as possible. We tried to respect the following criterions as we were increasing mesh density. Here are a few illustrations of what we obtained:



2) Grid effect conclusions and results

a) Final grid

We created three different grids, presenting an increasing number of nodes. The general repartition of the nodes, the growth rate factor of 1.1 from the airfoil to the outer boundaries is conserved. Lastly, the first height, concerning the wall adjacent cell does not require particular attention. This is due to the inviscid properties of the flow that implies $V_{wall} = V_{\infty}$. We fixed $\Delta s_{init} = 0.01$.

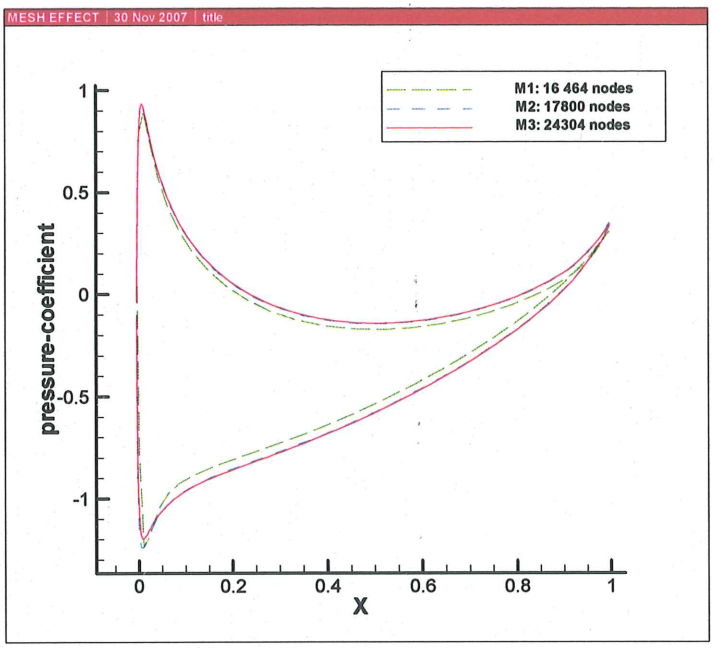


Figure 20: Mesh effect via the pressure coefficient distribution

On *figure 20* there a three pressure coefficient distributions, each corresponding to a different mesh. From M1 to M3, the number of nodes increases. The coarser mesh M1 (≈16 000 nodes) is not dense enough since a denser mesh M2 (≈17800 nodes) has an important effect on the results. The mesh M1 was underestimating all of the C_P values. However, the differences between M2 and M3 are negligible despite the 6500 additional nodes. We conclude that mesh independency has been reached and that mesh M2 is dense enough to accurately capture all the existing gradients.

b) Effect of the CFD domain dimensions

In this paragraph, the aim is to confirm that the numerical domain's dimensions have a noticeable effect on the lift coefficient for instance. The following table lists the values of C_L we obtained with *Fluent* for a series of domain sizes. The domain surrounding the airfoil was chosen to be circular. In part I of our study, the Karman-Trefftz transformation gave us all the distances the domain should have to insure that the relative error on the lift coefficient does not exceed 0.005. For all of the airfoils, the longest distance was always the upper vertical one, Y_{up}. Therefore we are only focusing on this distance which is why the domain is a circle. The theoretical results are all listed in *Appendix A_2/*, so we refer to those of the Karman-Trefftz airfoil KT_00 15 40 and more specifically the Y_{up} values. The CFD calculations were run at $\alpha = 5°$. Let us now comment on the tabulated results. The values of C_L are increasing with the distance L, so we deduce that the error is reducing and C_L getting closer and closer to a stable value. Indeed, from L = 25 on, C_L does not depend anymore on the domain's size. We may say that the results are *'size independent'*. As for the theoretical Karman-Trefftz value, we got: $\alpha \approx 5° \Rightarrow Y_{up\_KT} \approx 20$. We have no numerical data for Y_{up} equal to 20 so we cannot know how close the two results are from each other. However, both numerical and theoretical are in good agreement.

Dimension L (chord)	Lift coefficient
5	1.1533
10	1.169
15	1.1809
25	1.2076
35	1.2076

c) Comparison of pressure distributions

The numerical and analytical results are compared in this section. We obtained the pressure coefficient along the airfoil for several incidences: $0°, 5°, 10°$, which are represented on the three following plots:

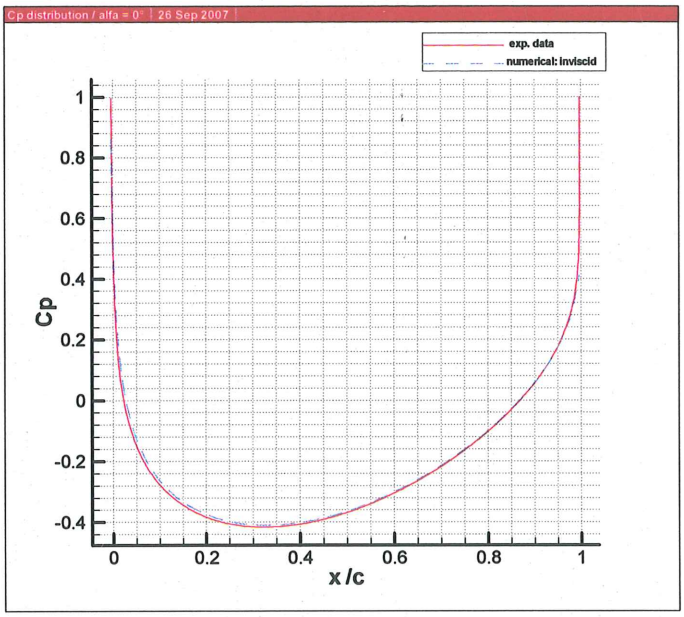


Figure 21: Cp at 0 degrees incidence

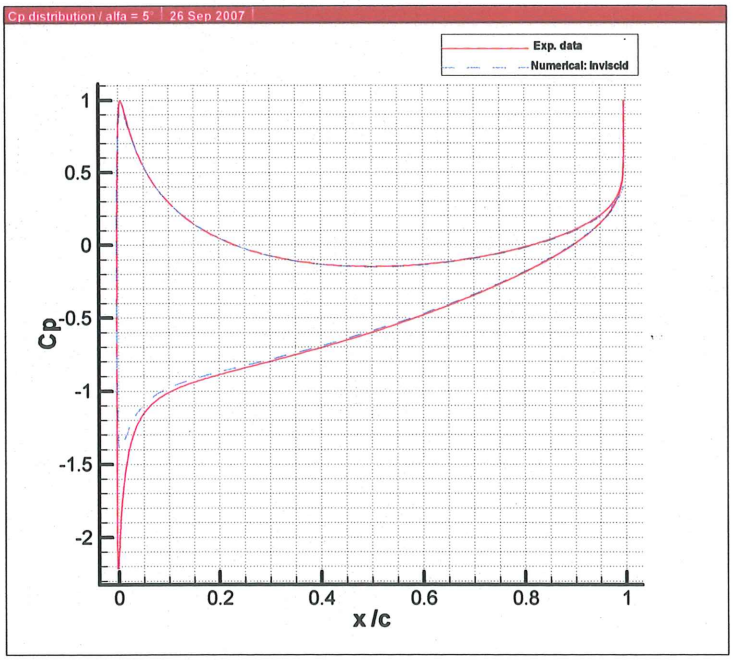


Figure 22: Cp at 5 degrees incidence

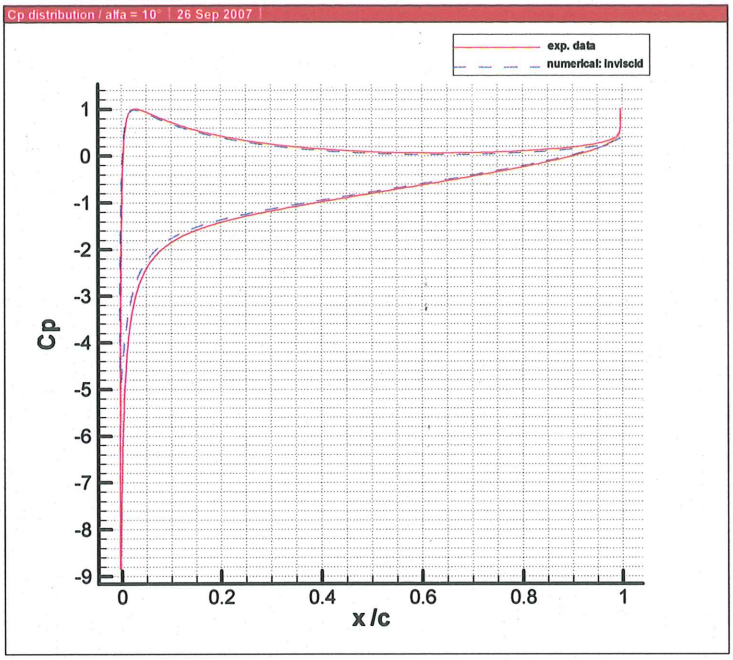


Figure 23: Cp at 10 degrees incidence

The numerical and analytical results appear to be almost the same. For the lower surface the difference is very small and the two curves are practically identical. This is the case for the three incidences. However, the pressure upon the upper surface differs a bit. The numerical simulation is slightly underestimating the values compared to the Karman-Trefftz results.

One of the major differences between the curves is the value of Cp towards the trailing edge. It is ideally equal to 1 since it is a stagnation point (velocity is zero). The Karman-Trefftz transformation, while respecting the Kutta condition automatically sets Cp to 1. However, for the numerical calculation, we would need an infinite number of points to reach it. As the number of nodes increases, we get closer and closer to the correct result. The same observation can be made concerning the peek at the front of the airfoil. The analytical results have a much larger peek at this point, which is caused by the drop in pressure due to the accelerating flow; once again it all depends on the number of points used for the CFD calculations. As this number increases, the numerical peak gets closer to the analytical value. However, the rest of the pressure distribution is grid independent long before we get close to Cp = 1 at the trailing edge. The distribution remains identical except for the value at the stagnation points. Therefore, a finer mesh seems unnecessary. For all three incidences, the mesh was fixed at 'grid-independence' without taking into account the stagnation point values.

d) Lift coefficient as a function of the incidence

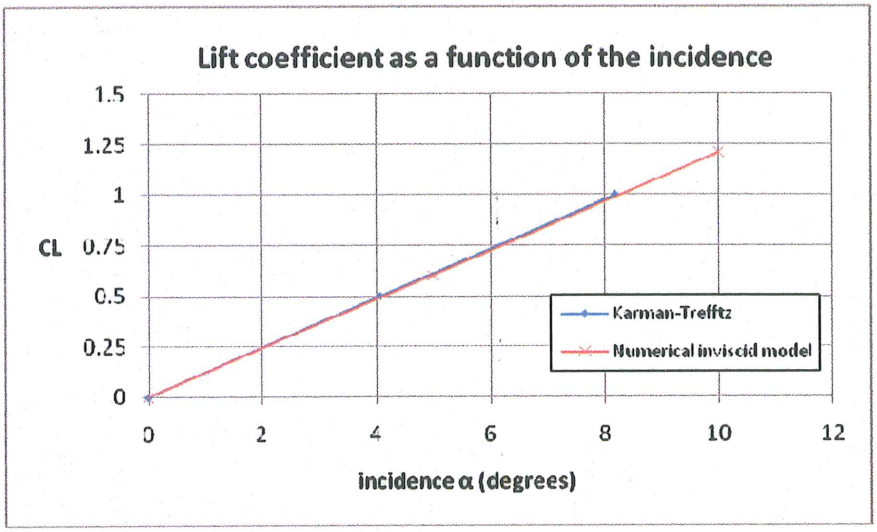


Figure 24: Evolution of C_L with respect to α

The lift coefficient is an important aerodynamic characteristic for an airfoil and is often related to the angle of attack of the flow. The resulting curve, which for a non viscous flow is a straight line since the flow stays fully attached to the surface, is the characterising trend of a specific airfoil. Thus, we decided to conclude this inviscid part of the project by comparing the theoretical and numerical linear fits. The results of *figure 24* appear to be satisfying since both lines are close to identical; the slopes are equal to 0.12 and 0.122 respectively which represents a relative error of only 1.6%.

III. VISCOUS CALCULATIONS

1) Introduction

Up to now the calculations and simulations we have made were using the potential flow theory. The effect of viscosity was excluded from the problem's formulation. The velocity field all around the airfoil's surface was calculated analytically by applying the Karman-Trefftz transformation; we consequently deduced the related pressure distribution. As for the numerical simulations, an inviscid model was used, allowing us to obtain the entire flow field within the computational domain after the Euler system of equations solved. It is shown that inviscid calculations do not accurately describe the observed pressure distribution over an airfoil section. Indeed, the viscous forces are not taken into account while their contribution is fundamental.

At low angles of attack, the potential flow theory is extremely accurate and provides precise estimates of lift and pressure. However, as the incidence increases, the results begin to deviate more and more up to a point where the lift coefficient suddenly drops. Indeed at high angles, the airfoil stalls as the flow separates, which cannot occur in the inviscid model, so that the potential flow curve of C_L as a function of the incidence continues to increase linearly. Indeed, the friction effects are neglected, and therefore the velocities at the surface are equal to those of the free stream.

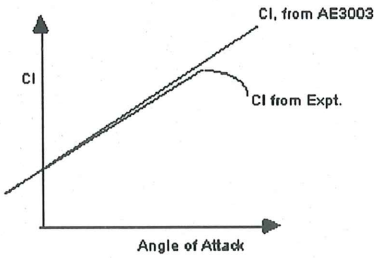


Figure 25: C_L versus angle of attack

In reality, when fluid particles come close to the surface their velocity slows down due to the viscous friction. Viscosity is a physical property that affects stresses of a fluid due to fluid motion. In the case of a viscous fluid flowing past a body, it adheres to the body surface and frictional forces retard a thin layer of fluid adjacent to the surface. The velocity then becomes a function of the distance from the surface and it is only at a certain distance that it is equal to the free-stream velocity. The distance (δ) required by the fluid to reach 99% of U_∞, is known as the boundary layer thickness.

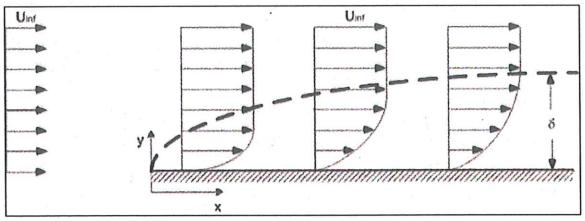


Figure 26: Boundary layer thickness δ

As a result, the velocity inside the boundary layer is less than the velocity at its outer edge. The existence of this velocity deficit is a necessary condition for separation. At the outer edge of the boundary layer viscous forces are negligible, and there is an exact balance between inertia and pressure gradient, as expressed by the Bernoulli equation. In the case of an airfoil, the curvature of the top surface caused by the angle of attack forces the flow to first accelerate around the leading edge and then decelerate. While the pressure increases as the particle moves downstream, it is accompanied by a decrease in velocity. The inertia of the particles near the wall

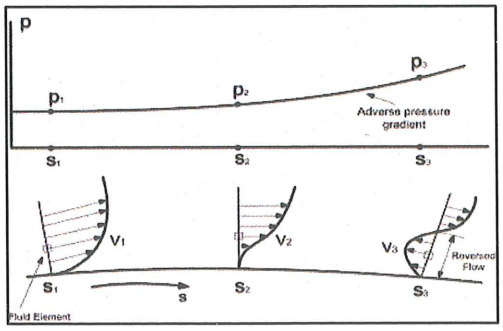


Figure 27: Reversed flow in the presence of an adverse pressure gradient

may not be sufficient to overtake the pressure forces, causing the velocity vector to change direction. This velocity deficit indicates separation. *Figure27* illustrates the different stages leading to separation. As we can see, the process can only be initiated by a sufficiently strong adverse pressure gradient. On the schema, it first increases up to an inflection point known as the separation point. The wall shear stress is exactly zero. As the unfavourable pressure gradient increases, the velocity gradient at the wall decreases and may become negative indicating the occurrence of reversed flow. The laminar or turbulent boundary layer has detached from the surface, resulting in a region of recirculating flow.

Now that the calculations we will run consider viscous flows, the separation problem will most certainly occur since the angle of attack will vary in between $[-12°; +24°]$. The objective is to develop a correct numerical prediction of the flow. Indeed, the viscous forces dramatically affect the drag coefficient of the airfoil so that an accurate description is essential. For closed 2D geometries, the potential flow was predicting zero drag at all angles of attack. However, any airfoil produces drag. Between the solid and fluid, the solid slows down the fluid flowing down over it, and shear stress develops, thus raising the skin friction drag. This value is proportional to the viscosity times the velocity gradient.

Our study is limited to the capabilities of *Fluent* in correctly predicting the 'real' flow. *Fluent* provides a wide range of modelling capabilities, for laminar and turbulent flow problems. The airfoil we will be studying is the 4-digit NACA 4412. This thoroughly tested airfoil provides us with a lot of experimental data. After deciding which turbulent model to use we will compare the resulting pressure distributions along the airfoil with the experimental results. Pressure distributions over an airfoil section directly provide the knowledge of the air force distribution along the chord that is required for some purposes. Depending on the flow conditions, in particular over a wide range of incidences, the aim is to understand why a particular model should be favoured with respect to the other.

2) NACA 4-digit airfoil

a) Presentation

Before the National Advisory Committee for Aeronautics (NACA) developed the NACA airfoil series, airfoil design was rather arbitrary. There was nothing to guide the designer except past experience with known shapes and experimentation with modifications to those shapes. These methods began to change when it was noticed that many similarities exist between the most successful airfoils. The two primary variables that affect those shapes are the slope of the airfoil's mean camber line and the thickness distribution above and below that line.

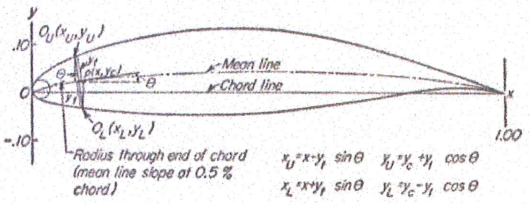


Figure 28: NACA airfoil geometrical construction

A series of equations incorporating these variables was presented and could be used to generate an entire family of related airfoil shapes. Despite the more sophisticated designs, those two basic geometrical values remained at the heart of all NACA airfoil series. The early NACA airfoil series

was generated using analytical equations that describe the curvature (camber) of the mean line, as well as the section's thickness distribution along the length of the airfoil. The NACA 4-digit series was the first family of airfoils using this approach. Let us specify what each digit represents for these airfoils:

→ **1st digit:** Maximum camber in percentage of the chord

→ **2nd digit:** Position of the maximum camber in tenths of chord

→ **3rd and 4th digit:** Maximum thickness of the airfoil in percentage of chord

b) NACA 4412

- Scale effect

The results of experimental pressure measurements of the NACA 4412 airfoil can be found in various publications. It directed our choice to study this particular airfoil and we will refer to the data collected by *'Robert M.Pinkerton'* in the variable density wind tunnel of the National Advisory Committee for Aeronautics (NACA). All the experimental results were found in his report published in 1937: *'Calculated and measured pressure distributions over the mid-span section of the NACA 4412 airfoil'*.

The new NACA variable density wind tunnel was very innovative back then and we will briefly explain what its purpose was. The aim was to eliminate the scale effect, inevitable when wind tunnel testing at a reduced scale is undertook. It introduces major errors which affect the accuracy of the results. The leading feature of this new design concerns the working fluid since the air under normal conditions is replaced by highly compressed air. But why do experimentalists think that they can learn what will occur on a large scale by observing what occurs on a small scale? For the boundaries of the flow to be close to identical, a constant ratio has to be conserved in between the characteristic quantities of the two arrangements. Thus, this principle needs to be applied not only to the model's dimensions but to the flow properties as well. If the flow is not exactly similar but only approximately, the information becomes unreliable. A 'scale effect' exists and must not be underestimated. The first assumption was to assume that none of the physical properties of the fluids has any influence on the shape of the flow pattern or on the fluid forces, despite the density of the fluids. Therefore, the mass force of the particles is the only force and needs to be equalized. Indeed, for a viscous flow, it is arranged so that the pressure forces and viscous forces are in equilibrium with the mass force. To obtain the criterion for the similarity of flows, two of the three forces need to be changed by an identical ratio (mass and viscosity forces), in order to maintain equilibrium. The mass forces are changed in the ratio $\frac{\rho_2 V_2^2 l_2^2}{\rho_1 V_1^2 l_1^2}$ and the viscous forces with $\frac{\mu_2 V_2 l_2}{\mu_1 V_1 l_1}$. Hence the condition for an exact model test is: $\left(\frac{\rho_2 V_2^2 l_2^2}{\rho_1 V_1^2 l_1^2} = \frac{\mu_2 V_2 l_2}{\mu_1 V_1 l_1}\right) \Rightarrow \frac{V_1 l_1 \rho_1}{\mu_1} = \frac{V_2 l_2 \rho_2}{\mu_2} \Leftrightarrow \mathrm{Re}_1 = \mathrm{Re}_2$.

The equality of the two Reynolds numbers ensures the dynamic similarity of the flows.

It is only if in addition the two bodies are geometrically similar that the scale effect is perfect. To conclude, solving the scale effect problem is the main difference between this wind tunnel and the others and therefore eliminates the most serious source of error. Usually, the model dimensions are much smaller than the original of course, which requires either a large increase in velocity for $V \times l$ to remain constant or the diminution of μ/ρ. In the first case, the velocity increase generates excessive

air forces on the model accompanied by a pressure difference leading to greater compressible effects than desired. In the second case, the choice of another fluid is the only possibility but it needs to be denser than air and have a smaller dynamic viscosity than air. However no such fluid exists. To conclude, a classical wind tunnel operating with air under standard conditions is unable to fully satisfy the scale effect conditions. It is only by changing the density of the air that similarity is obtained.

- NACA wind tunnel

In this paragraph, a brief description of the wind tunnel and the methods of operation are given. The next figure illustrates the NACA variable density wind tunnel.

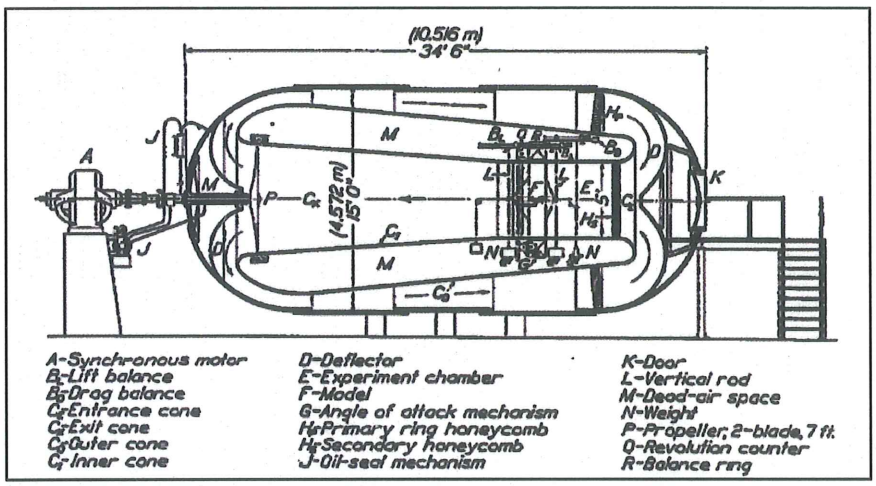


Figure 29: Sectional elevation of variable density wind tunnel

It consists in an experimental section (E), 1.52 meters in diameter, where the entrance and exit cones are housed in a 4.57 meters diameter steel tank, of 10.52 meters long. The balance is located in the dead air space between the walls of the experimental section and the outer case. The air is circulated by a two blade propeller. In an ordinary atmospheric wind tunnel, variations in dynamic pressure are possible by changing the air velocity. Here, it is done by changing the density of the air so that any velocity variation is unnecessary. The air is compressed in two or three stages according to the final pressure of the tank. For pressures up to about 7 atmospheres, a two stage primary compressor is used. For higher pressures, an additional booster compressor is required.

During the data acquisition, pressures were simultaneously measured at 54 orifices, $2.03*10^{-4}$ meters in diameter, over the mid-span section of the NACA 4412 airfoil, for 17 angles of attack within the range $[-20°; +30°]$. The model used was a standard duralumin airfoil composed of the NACA 4412 section, a rectangular span of 0.762 meters and a chord of 0.127 meters. One of the mid-span sections was replaced by a brass section in which the orifices were located. They were placed in 2 rows and drilled perpendicularly into the airfoil as follow:

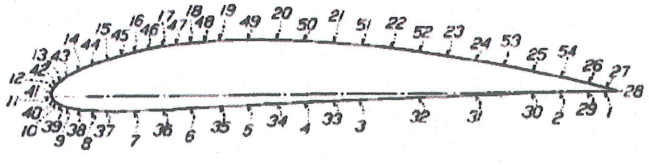


Figure 30: Distribution of pressure orifices along the NACA 4412

A large number of orifices were positioned towards the leading edge in order to evaluate correctly the pressure distribution in that region of strong fluctuations. Brass tubes were fitted to the holes and brought out of the model to the dead air space. In that area, they were connected to a photo-recording multiple-tube manometer. The reference pressure was measured by a pitot head. The way of determining the pressure data is direct and simultaneous, achieved by recording of the liquid's height in the manometer tubes.

c) Creation of the geometry and CFD domain

- Geometry

The experimental data concerning the geometry's coordinates was insufficient. Only 20 points were reported, which once linked would produce a rough contour. It is essential for us to generate a smooth geometry so that the pressure distribution is unaffected by any approximations or abrupt curves. Consequently we calculated the coordinate distribution with a simple *Matlab* program *(ref. Annexe D.)*

The coordinates are based on the mean camber line and the thickness distribution of the profile. As we can see on the schema, the mean camber line is drawn half way between the upper and lower surfaces. The ratio thickness to chord and the maximum camber are needed.

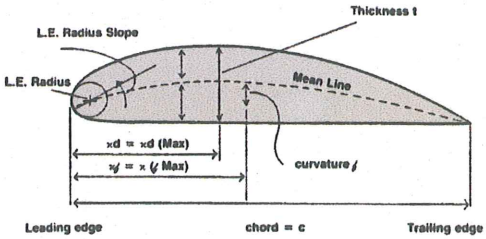


Figure 31: Geometrical properties of an airfoil

Here is how the coordinates are expressed:

$$\begin{cases} y_{camb}/c = \dfrac{2x_1 f}{(x_1 c)^2} \times (x/c) - (x/c)^2, & (x/c) \le x1 \\ y_{camb}/c = \dfrac{f}{c(1-x_1)^2} \times \left[(1-2x_1) + 2x_1 \times (x/c) - (x/c)^2 \right], & (x/c) > x1 \end{cases}$$

$$\frac{y_{thick}}{c} = 5t \times (0.29690x^{1/2} - 0.12600x - 0.35160x^2 + 0.28430x^3 - 0.1015x^4)$$

$$with: \begin{cases} f : max.\,camber\ (position \rightarrow xf) \\ x_1 = xf\,/\,chord \\ t : thickness\,/\,chord \end{cases}$$

We applied the following formulas to calculate the coordinates of the upper and lower surfaces. We did not neglect the angle θ shown on *figure 28* as it was sufficiently affecting the geometry to impact on the results:

$$\begin{cases} x_{upper} = x - y_{thick}.\sin\theta & y_{upper} = y_{camb} + y_{thick}.\cos\theta \\ x_{lower} = x + y_{thick}.\sin\theta & y_{lower} = y_{camb} - y_{thick}.\cos\theta \end{cases}$$

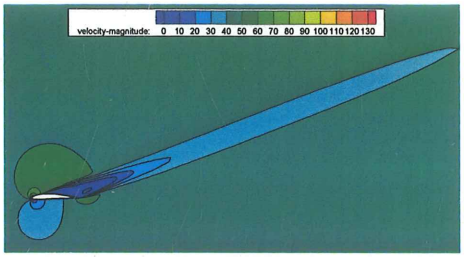
- CFD domain

The boundaries of the CFD domain have to be carefully defined because if there are too close it may affect the results. The inviscid study we undertook previously provided supporting evidence to this fact. For the domain to have a negligible impact, the distances had to be at least equal to the values calculated by the Karman-Trefftz theory. For values under these limits, the numerical calculations showed that the smaller the distances the larger the error on the flow variables.

For this viscous study, the Karman-Trefftz calculations can be directly applied. Although they were obtained for inviscid conditions, the influence of the circulation is identical. Viscosity only affects the CFD domain close to the airfoil, so that the velocity contours are very different. However, what influences the domain further away from the airfoil is the circulation and the CFD boundaries depend entirely on this amount. Consequently, all the results we obtained in part I can be applied directly for any viscous study. In part III, the objective is to evaluate the reliability of two different turbulence models. We chose to refer ourselves to past numerical computations in order to fix unique CFD

boundaries. Indeed, if we rely on the Karman-Trefftz results, the domain would have to be modified for every incidence. Therefore, based on very reliable sources that have proved to be valid for many numerical studies, this simplification can only affect our CFD results in a negligible way. These sources are numerous of course and it quickly appeared that the outlet had to be at a largest distance from the airfoil than the inlet. Indeed, if the flow separates the impact is much greater downstream. The velocity contours on the right were obtained from one of our simulations and illustrates a typical contour pattern for α larger than 20 degrees. Our references indicated that the following distances are sufficiently far not to affect the results:

Velocity contours around an airfoil at high incidences



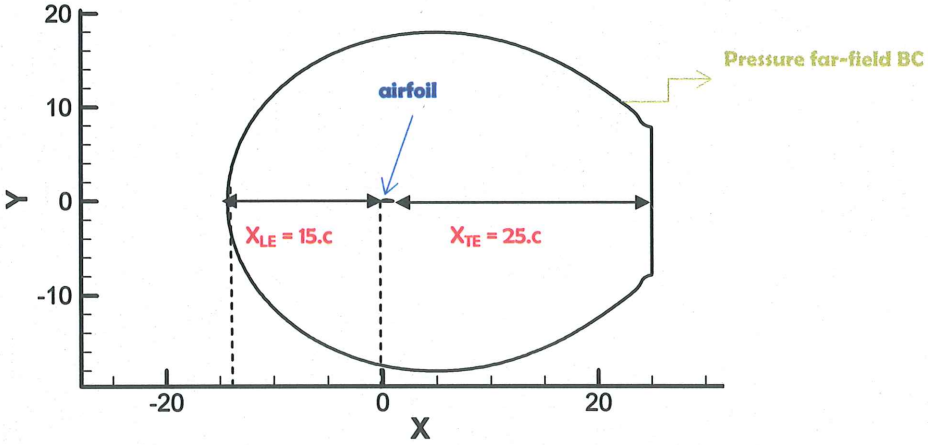


Figure 32: Domain dimensions

These have to be carefully respected. We will apply a pressure far field boundary conditions to the entire outer domain. If the boundaries are too close to the airfoil, the free stream conditions we define as parameters of the BC do not apply yet, in reality they do but further away. This approximation would introduce a certain error and lead to inaccurate results.

3) Grid generation around the airfoil

a) Presentation of Gridgen

For the generation of the mesh surrounding the NACA 4412 airfoil, we decided to use the software *Gridgen* instead of *Gambit.*

Since 1984, *Gridgen* has been used to create three dimensional grids for complex geometries in a production environment. The software origins are in the demanding US aerospace industry where it has a great reputation. It is known for creating high quality grids which is vital for reliable simulations. Due to the enhanced quality of the mesh, it leads to more accurate solutions and faster convergence. Now days, the software is used worldwide in aerospace, automotive, power generation, chemical process and many other industries for which CFD is an integral part of the design process.

This part of our study includes viscous effects so the quality of the mesh especially close to the airfoil surface becomes increasingly important. Indeed for the inviscid study, the potential flow did not require to refine the mesh in that region. Mesh quality is where CFD has the largest impact on solution accuracy, so particular attention was given to it from the start of our simulations, first by the

choice of the meshing software. Just as *Gambit* does, *Gridgen* provides several diagnostic tools to ensure that mesh quality criteria is met. Cell size, cell-to-cell size variation, aspect ratio, Jacobian, skewness, smoothness...etc, all of these can be visualized graphically. However, the very interesting feature that *Gridgen* possesses but not *Gambit* is that these quality diagnostics are integrated in the meshing process so quality may be monitored as the mesh is being generated.

b) Extrusion process

The major difference between both softwares is the way the mesh is generated. In *Gambit*, the domain boundaries first had to be defined, thus creating a closed face, and finally the empty space would be filled by the mesh. On the other hand, *Gridgen* never requires outer boundaries to be created since the mesh is extruded directly from the airfoil surface.

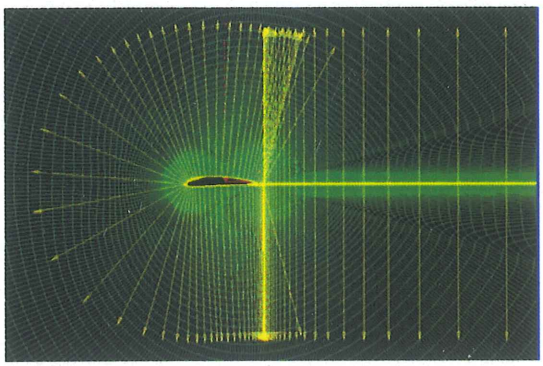


Figure 33: Gridgen's extrusion tool

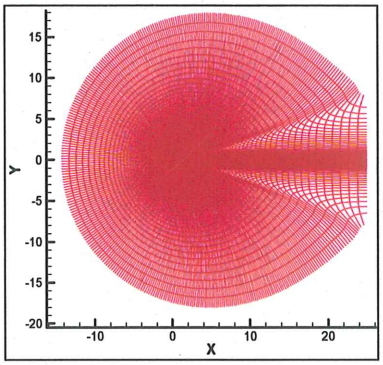


Figure 34: Final CFD domain after extrusion

As we can see on *figure 33*, to define the single edge we started by the downstream edge linked to the trailing edge of the airfoil, selected the upper and lower surfaces of the profile and again the downstream edge. For the extrusion process, the first height of the mesh, the growth rate and the number of steps have to be specified.

c) Elliptic solver

The quality of structured grids can be significantly improved by applying *Gridgen*'s elliptic PDE methods. As we can see on the two figures below, the elliptic solver modifies the mesh so that overall, it is much more orthogonal to the surface from which the extrusion originates. *Figures 35 & 36* are focusing on the upper surface of the aircraft.

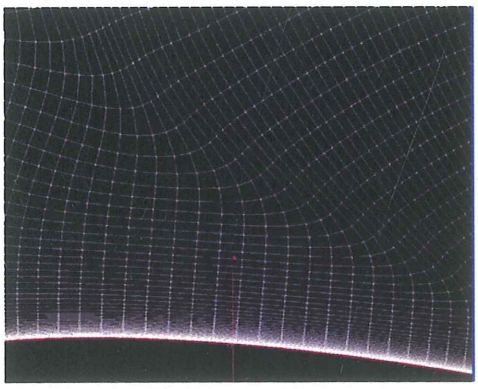


Figure 35: Initial mesh just after extrusion

After 2 iterations

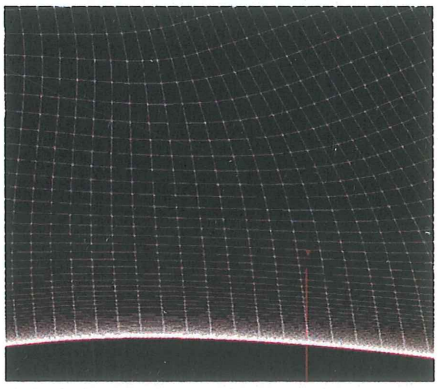


Figure 36: Resulting mesh after the elliptic solver

It is a critical region since at higher angles of attack the flow will separate and this area will be the most challenging to solve due to the high turbulence levels. Initially, some of the cells were quite distorted and of bad quality as we observe on *figure 35*. Running the solver for a couple of iterations allowed the mesh to globally straighten up and the resulting mesh is of much better quality. We will understand later the importance of well capturing the flow variables in the separation region. Indeed, the aerodynamic characteristics we wish to obtain are highly dependent on the amount of separation.

d) Growth rate

Concerning the growth rate we applied, the best mesh was obtained by combining several values. Once again, our major concern is the region where separation is most likely to occur. *Figure 36* just above was used to highlight the elliptic solver's advantages but in reality the cells are too large.

Indeed, at higher angles of attack the flow contains very small gradients and they are not captured accurately. We have run several simulations for separated flows with such a grid and the results were not satisfactory. This important deviation from the experimental results was corrected by refining the region. To do so, the usual growth rate of 1.1 particularly adapted to aircraft applications, was not applied so soon in the extrusion. The process was divided as follow:

	Number of extrusion steps	Growth rate
Near to wall region until a reasonable cell is obtained	30	1.1
This last cell was slowly increase throughout the separation region	50	1.02
Average growth rate to link separation and far regions	10	1.06
The rest of the space is filled with the usual growth rate	40	1.1

As we can see on *figure 37*, the mesh we obtain from these multiple growth rates is much finer than the grid of *figure 36*. This operation was necessary because we will be using the same mesh all across the incidence range. It is a loss of memory and time for the incidences where the flow remains attached but is necessary these higher angles of attack.

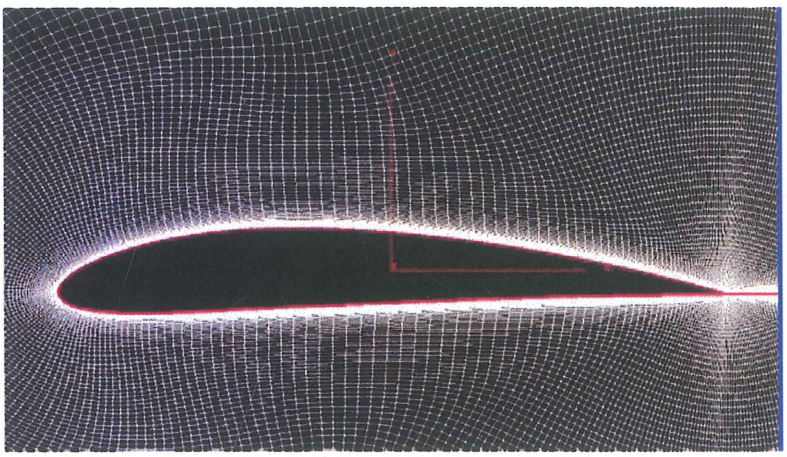


Figure 37: Refinement of the separation region

4) Turbulence modeling

a) Introduction

Almost all fluid flow which we encounter in daily life is turbulent. The flow around bluff bodies such as cars, aeroplanes, buildings...etc, are turbulent. The flow within combustion engines, in both piston and gas turbine engines as well as combustors, is highly turbulent. Air movements in rooms are also turbulent, at least along the walls where wall-jets are formed. Thus, while computing the fluid flow, it will almost always appear to be turbulent. In our study, we will be analysing the flow past an airfoil. Of course, turbulence is also very likely to occur. The flow over the bulk of the wings and fuselage is usually turbulent in the boundary layer region where the velocity of the airflow changes to match the velocity of the aircraft. There can be complicated leading edge effects, where turbulence usually manifests itself. Turbulence flow over the wings will be subjected to variable pressure gradient and ultimately the turbulence will change character at the trailing edge as the flow changes from the attached boundary layers to a free wake. Downstream of the aircraft the turbulent trailing vortices decay only very slowly, which determines the minimum aircraft spacing required for safe flight. Turbulence is also responsible for a significant proportion of the sound produced by an aircraft.

So what is the nature of turbulence? It can first of all be considered as a broadband existing over a wide range of spatial and temporal scales. Turbulence is not at all a bad thing for many applications. For instance, it is helpful in most combustion engines since it efficiently mixes the fuel and the oxidiser. It is also much better able to resist boundary layer separation and wing stall in aircraft applications. However, it comes with a high friction drag penalty.

In turbulent flows, the variables are usually divided in two. $\overline{U}$ represents the time-averaged part and is independent of time for a steady mean flow, whereas u is the fluctuating part. Therefore, the variables are expressed as follow: $U = \overline{U} + u$

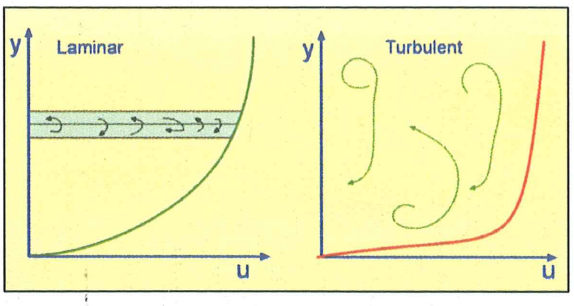
Turbulent flows have a certain number of characteristics that we can briefly describe.

First of all, turbulent flows are '*irregular*', random and chaotic. The flow consists in a spectrum of different scales known as eddy sizes. The largest eddies are of the order of the flow's geometry, for instance the boundary layer thickness with length scale l and velocity scale U. The scales extract kinetic energy from the mean flow. The large scales interact with slightly smaller scales to which the kinetic energy is lost. It is through this cascade process that kinetic energy is transferred from the larger scales to the smaller scales. The smallest scales where dissipation occurs are called Kolmogorov scales: The friction forces become larger and most of the kinetic energy transferred from the larger scales is finally dissipated into internal energy. The characteristic scales of these small eddies are the velocity scale v, the length scale l and the time scale τ. Since the kinetic energy is destroyed by the viscous forces, we naturally suppose that viscosity plays a part in determining these scales. Also we know that the amount of energy to be dissipated is ε. Therefore, we assume that these scales are determined by both viscosity ν and dissipation ε. The amounts v, l *and* τ are expressed according to viscosity and dissipation thanks to a dimensional analysis:

$$\begin{matrix} v & = & \nu^a & \varepsilon^b \\ [m/s] & = & [m^2/s] & [m^2/s^3] \end{matrix} :$$

The '*diffusivity*' of turbulent flows increases. Consequently, the spreading rate of boundary layers, jets, etc. increases as the flow becomes turbulent. As we observe the two velocity profiles on

the right, we see that for a laminar boundary layer, the flow takes place in layers. Exchanges of mass or momentum only occur between adjacent layers on a microscopic scale. On the other hand a turbulent boundary layer is marked by mixing across the layers, now occurring at a macroscopic scale. Consequently, it has a steep gradient of velocity at the wall. If the flow experiences adverse pressure gradient at the wall it will cause the flow to slow down. In the case of a turbulent boundary layer, separation is delayed since the velocities are higher at the wall. The velocity distribution takes longer to reach the inflection point, that characterizes the transition to separation.



Turbulent flow occurs at *'high Reynolds numbers'*. For example the transition to turbulent flows in pipes occurs when $\mathrm{Re}_D \simeq 2300$, and in boundary layers at $\mathrm{Re}_x \simeq 100000$.

Turbulent flow is always '*three-dimensional*'. However, simplification to two-dimensions can be very helpful for the understanding of certain problems, but does remove some important physical mechanisms. It is usually recognized as significantly different from the full turbulence problem.

Turbulent flows also have a '*dissipative*' aspect as we briefly introduced while presenting the large and smaller scales. It causes the kinetic energy to transfer from the large eddies to the small eddies. This process of transferred energy from the largest turbulent scales to the smallest is called cascade process.

Finally, even though turbulent flow is composed of small scales as well, they are much larger than the molecular scale and we can treat the flow as '*continuum*'.

b) Turbulence models

- Introduction

It is necessary to decompose the instantaneous variables into a mean value and a fluctuating value. For example, the pressures and velocities are rewritten as:

$$U_i = \overline{U_i} + u_i$$

$$P = \overline{P} + p$$

Indeed, this decomposition is useful because while measuring the flow, we are interested in the mean values of these variables and not their time history. For numerical resolutions, if the flow was always to be considered as time dependant, which is always the case since turbulent flows are unsteady by nature, it will require a very fine resolution in time. Therefore, when experimental and numerical results are compared, it is the mean values of pressure, temperature..., etc. that we compare. For instance, the pressure probes' computerized system is automatically averaging the values in time, just like the steady numerical simulation is doing. To solve the flow, the continuity equation and the Navier-Stokes equation are considered:

$$\frac{\partial \rho}{\partial t} + (\rho U_i)_{,i} = 0 \qquad (1)$$

$$\frac{\partial \rho U_i}{\partial t} + (\rho U_i U_j)_{,j} = -P_{,i} + [\mu(U_{i,j} + U_{j,i} - \frac{2}{3}\delta_{ij} U_{k,k})]_{,j} \qquad (2)$$

As far as the notation are concerned, *',i'* and *',j'* indices refer to the term's derivative with respect to x_i and x_j respectfully. The next step is to replace the decomposed variables into equation *(1)* and *(2)*.

We make the assumption that the flow is incompressible, meaning that density does not depend on pressure. We obtain the time averaged continuity and Navier-Stockes equations:

$$\frac{\partial \rho}{\partial t} + (\rho \overline{U}_{,i})_{,i} = 0 \qquad (3)$$

$$\frac{\partial \rho \overline{U}_{,i}}{\partial t} + (\rho \overline{U}_{,i} \overline{U}_{,j})_{,j} = -\overline{P}_{,i} + [\mu(\overline{U}_{i,j} + \overline{U}_{j,i} - \rho \overline{u_i u_j})]_{,j} \qquad (4)$$

The dilatation term of equation *(2)* was neglected due to incompressible effects and an additional stress term $\overline{u_i u_j}$ appears in *(4)* due to turbulence, created by the fluctuating velocities existing within the flow. This term called the *Reynolds stress tensor* is unknown and needs to be determined to close the equation system. Indeed we have ten unknowns (3 velocity components, 6 stresses and the pressure) for only four equations (continuity equation and the 3 components of the Navier-Stockes equation). This is known as the *closure problem*. To close the equation system, several levels of approximations can be used, thus generating different types of turbulent models. In our future numerical simulations we will only compare two turbulence models but let us briefly present the existing possibilities.

Zero equation models: For these models an assumption is needed to relate the Reynolds stresses to the velocity gradients via the turbulent viscosity. This relation is called the *Boussinesq assumption* and allows us to replace the Reynolds stress tensor by the product of the turbulent viscosity times the velocity gradients. The turbulent viscosity is often referred to as *eddy viscosity* and leads to *eddy viscosity models*. The particularity of these models is that they do not require the solution of any additional equation and are calculated directly from the flow variables.
These models are very useful for simple flow geometries or initial phases of a computation but are too simple for general situations. They are not able to account properly for convection and diffusion of turbulent energy.

One equation models: For these models, a transport equation is solved for a turbulent quantity which is usually the turbulent kinetic energy. The equation for the turbulent kinetic energy $k = 1/2\,\overline{u_i u_i}$ is derived from the Navier Stockes equation after making several simplifications which assume that the viscosity is steady, incompressible and constant. A second turbulent quantity is obtained and usually it is the unknown turbulent length scale that is needed. The length that is chosen usually depends on the situation. As for the turbulent viscosity it is again calculated thanks to the Boussinesq assumption.

Two equation models: For this type of model, two transport equations are solved which describe the transport of two scalars, usually the turbulent kinetic energy k and its dissipation ε. The eddy viscosity is obtained from k and ε. Finally, the Reynolds stress tensor is again obtained by assuming a certain assumption relating the tensor to the velocity gradients and the eddy viscosity.

Reynolds stress models: Up to now, eddy-viscosity based models have been dominating in the context of industrial flow computations. However, the requested degree of accuracy and the flow complexity are both becoming increasingly challenging. For example situations where separation is combined with flow control in highly curved surfaces need more complex models. Standard eddy-viscosity models usually under predict separation tendency. Indeed, the Boussinesq assumption appears to be rather crude. For Reynolds stress models, the aim is to remove the linear Boussinesq hypothesis, and replace it with a more general anisotropy relation. Up to then, the production model of the eddy-viscosity based models was insensitive to system rotation. This new and more precise

assumption allows the Reynolds stress models to get much closer to the flow physics occurring in reality. The method to close the set of equations is similar. One transport equation is derived and the Reynolds stress tensor and another is necessary for determining the length scale of the turbulence.

The order in which we have listed these different types of turbulence models has its importance since they are becoming more and more complex and expensive in terms of computational time. After taking these considerations into account, we decided to compare two of the two equation models, the k-ε Realizable Model and the k-ω SST Model. They are by far the most popular turbulence models utilized now days because they offer a good balance between complexity and accuracy.

- The k-ε Model

For the k-ε model, as its name indicates, the two transport equations are solved for k the turbulent kinetic energy and ε its dissipation. The turbulent length scale obtained is equal to: $l = \frac{k^{3/2}}{\varepsilon}$.

As for the turbulent viscosity, it is related to k and ε as follow: $\nu_t = c_\mu \frac{k^2}{\varepsilon}$.

The unknown constants are determined thanks to simple flows where the equations can be simplified. They are then determined by using experimental data.

The **k-ε Realizable** model is very similar to the standard k-ε model but has an additional constant C_μ and a modified ε equation. The main advantage of this Realizable version is that it works much better that the classical formulation in regions which experience strong accelerations and decelerations. It is particularly adapted to our case since we will be studying a high Reynolds flow, going through an important deceleration at the leading edge and acceleration in the region around the suction side where the pressure is at its minimum. The model is a high-Re model so it has problems in correctly treating the flow close to the wall. Indeed, a low-Re number model is needed to complement the model for solving the boundary layer if a grid with resolved boundary layers is used.

- The k-ω Model

Just like for the k-ε model, the turbulent kinetic energy k is solved. However, instead of using the dissipation ε to determine the turbulent length scale, a *specific dissipation* term ω, proportional to the ratio of k over ε, is used. This modification in the dissipation term avoids the k-ω model to encounter problems when $k \to 0$. Indeed, the k-ε model develops large numerical problems in this situation within the ε equation, which do not occur for the ω equation. However, the model appears to be too sensitive to inlet free-stream turbulence properties.

The **SST k-ω** turbulence model was implemented by F. R. *Menter* in 1993. It is now days becoming very popular. The model is a combination of the k-ε and k-ω models we just presented. The k-ω formulation is used in the inner parts of the boundary layer so that it can be used directly all the way down to the wall. This extends the model capabilities to low-Re (wall) applications. The SST model switches to the k-ε model behaviour in the free-stream. Therefore, it does not inherit of the k-ω disadvantage of being too sensitive to the free-stream. To conclude, it seems like the SST model combines the best of two worlds but it does present a major drawback. It needs to calculate the distance from the wall. For an unsteady simulation, the calculation becomes extremely expensive

since the height is determined at each time step. The problem is even worse for unstructured grids because the value varies not only in time but spatially as well. However, we will be making steady calculations on a structured grid; therefore we will not be penalized by this negative aspect. In addition, the model is known to behave extremely well in regions of adverse pressure gradients and separating flow.

c) Wall treatment

The numerical approach we use close to the wall is of major importance because large gradients in temperature and velocity occur in that region. Consequently, walls are the main source of turbulence. Near them, the flow is fully turbulent; while further away the turbulence is increasingly intermittent. The boundary layer can be divided into three layers:

- The viscous sublayer: at the wall the turbulent fluctuations have to disappear completely so that the no slip boundary condition is satisfied. Therefore, in a region very close to the wall called the *viscous sublayer*, the fluctuations are assumed to be very small and the time averaged flow must approximately respect this:

$$\nu \frac{d^2\bar{u}}{dy^2} \approx 0 \quad , \quad \textit{after integration}: \quad \bar{u} = \nu \left(\frac{d\bar{u}}{dy} \right)_w \qquad (1)$$

We will need a variable depending on the velocity gradient and the viscosity that, it is called the friction velocity and is defined as follow: $u_\tau = \sqrt{\nu \left(\frac{d\bar{u}}{dy} \right)_w}$

We can now define a dimensionless velocity u^+ and a distance normal to the wall y^+ according to the key variables affecting the close to the wall flow behaviour. Finally, the mean velocity expressed in (1) is rewritten: $u^+ = \frac{\bar{u}}{u_\tau}$ *and* $y^+ = \frac{y u_\tau}{\nu} \quad \Rightarrow \quad u^+ = y^+$

This approximation can be applied with good accuracy as long as $y^+ < 8$.

In the viscous sublayer, molecular viscosity makes the flow behave close to laminar.

- The buffer layer: *Prandtl* showed that in this inner region known as the buffer region, viscosity and friction velocity are still at the same scale than the turbulence. Indeed, the wall is still quite close and therefore the size of the turbulent eddies is limited. Consequently, a direct relation between the two dimensionless variables still exists. Just like previously for the viscous sublayer, a law could be determined experimentally, and appears to be a good fit to this delimited region:

$$u^+ = f(y^+) \quad \Leftrightarrow \quad u^+ = \frac{1}{\kappa} \ln y^+ + b \qquad 0.2\delta > y^+ > 30 \,, \quad \begin{cases} \kappa : \textit{von Karman constant} \\ b : \textit{additive constant} \\ \delta : \textit{boundary layer thickness} \end{cases}$$

The exact values of the constants are still being debated, in particular for high-Re applications, the most common values are: $\kappa = 0.41$ *and* $b = 5.2$

Within the buffer layer, the laminar and turbulent properties of the flow are both important. The following figure represents the behaviour of u^+ as a function of y^+, according to the three layers:

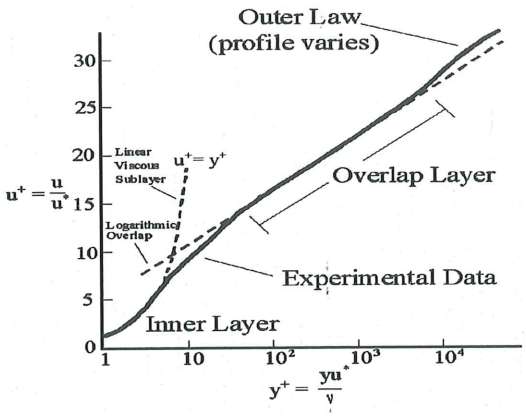
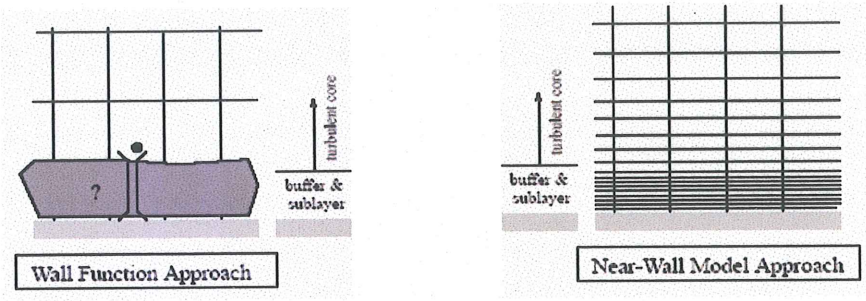


Figure 38: Decomposition of the boundary layer into three layers

- The outer layer: the outer region of the boundary layer has an edge velocity equal to U. An outer law for this region has also been developed and corresponds to this relation: $\frac{\bar{u}-U}{u_\tau} = g(\frac{y}{\delta})$. The outer layer is fully turbulent and the turbulent properties play the major role.

They are two approaches to solving the near wall problem:

1. Wall function approach, the flow is not solved but given by a function called *wall function.*
2. Enhanced wall treatment approach, the mesh is very fine and the flow solved everywhere.



1/ WALL FUNCTIONS

The wall functions are semi-empirical functions. Laws at the wall are used for mean velocities and scalars whereas formulae are more appropriated for evaluating the turbulent quantities. The link between the near-wall-cell solution variables and the respective quantities at the wall are respected.
In the *Fluent* software, two different wall functions exist.
The *standard wall function* utilizes a linear law for the mean velocity close to the wall, which then becomes logarithmic further away. As for the turbulence, in agreement with our description of the k-ε model, the standard wall function solves the k equation in the entire domain.
The *Non-equilibrium wall function* is particularly recommended for complex flows, where important pressure gradients, rapid changes...etc, occur. Indeed, the log-law used to determine the mean velocity is sensitized to pressure gradient effects. A two layer base allows for computation of turbulent kinetic energy in wall adjacent cells, so that viscous and turbulent layers may be treated correctly.

However, the wall function approach is rather limited. It will not be able to accurately solve the flow in many cases. For instance, flows with near-wall effects like a highly laminar flow, flows near rotating bodies implying the presence of strong body forces.... More importantly, we will be studying the flow past an airfoil for a large variation of incidences. Therefore, there will be strong pressure changes, in particular on the suction side of the profile where the variations will most probably not be accurately detected if the simulation is a wall function is implemented. Let us now present the second alternative which is to carry out an Enhanced wall treatment.

2/ ENHANCED WALL TREATMENT

In order to correctly resolve the viscous layer, the turbulence models need to be modified. An enhanced wall treatment will be computationally more costly since it is accompanied by a considerable refinement of the mesh. The mesh needs to be much closer to the wall so that the modified turbulent model is implanted right into the viscous sub-layer of the boundary layer.

<u>k-ε model</u>

The **enhanced wall treatment** option is **enabled**. For a fine mesh which will be our case, a two layer approach is applied. The near wall region is solved all the way to the viscous sub-layer. Within the viscosity affected region a one model equation of Wolfstein is solved whereas in the region slightly further from the wall which is turbulent, the k-ε model is applied.

<u>k-ω SST</u>

This model is available for both low and high Reynolds number models. The k equation has the same wall boundary conditions for the k-ω model than for the k-ε model when the enhanced wall treatment option is activated. For a coarse mesh a wall function approach is used, for which a logarithmic law of the wall is applied. In this case, the transitional flow option is not active. We want the boundary layer to be solved much more precisely; therefore the **transitional flow option** will be **enabled**. As a result, the wall shear stress is obtained from the laminar stress-strain relationship. The corresponding mesh guidelines for this option are to considerably refine the boundary layer mesh. These requirements are identical than for the k-ε model, since the first cell has to be contained in the viscous sub-layer, that is:

$$\boxed{\textit{At the wall adjacent cell} \rightarrow y^+ \approx 1}$$

However, a higher y+ is acceptable as long as it is well inside the viscous sub-layer: $y^+ < 5$

5) Results

a) Simulation parameters

We have decided to use the pressure far-field boundary condition proposed by *Fluent*. It models a free stream condition at infinity, for which we need to specify the free stream Mach number and static conditions. Therefore, as a preliminary stage before any simulation can be run, we have to calculate the far stream conditions corresponding to the experimental conditions.
We should specify that the pressure far-field boundary condition is only applicable when the density is calculated by using an ideal-gas law. Consequently, our calculation is based on two relations. First of all, the ideal-gas law that is governing the flow during the simulation. Secondly, the dynamic similarity criterion insures that the numerical and experimental conditions are equivalent.

- Ideal gas: $PV = nRT$

After manipulating the equation of state, we get a relation that depends on the pressure and density of the free stream such as:

$$P = \rho R_S T, \qquad R_S : Gas\ constant\ of\ a\ specific\ gas$$
$$R_{S\_air} = 286.9\ J/kg.K$$

We can deduce the density of the free-stream flow that is needed while manipulating the dynamic similarity. We choose to apply international standard atmospheric conditions, the result is:

$$\left.\begin{array}{l} T_\infty = 288.15\ K \\ P_\infty = 101325\ Pa \end{array}\right\} \Rightarrow \quad \rho_\infty = 1.225063\ kg/m^3$$

- Dynamic similarity: $\mathrm{Re}_{exp} = \mathrm{Re}_{num}$, $\mathrm{Re}_{exp} = 3100000$

From the expression of the Reynolds number, we directly deduce the free-stream velocity that we will have to impose: $\mathrm{Re}_{exp} = \dfrac{\rho_\infty U_\infty c}{\mu} \Rightarrow U_\infty = \dfrac{\mathrm{Re}_{exp}\,\mu}{\rho_\infty c} \Rightarrow U_\infty = 44{,}87675 \quad m/s$

However, the calculations are not totally finished since it is the free stream Mach number that is required, the final step is to deduce its value from the free-stream velocity and the speed of sound:

$$M = \frac{V}{a}, \quad where: a = \sqrt{\gamma R_S T} \Rightarrow M = 0.131927$$

To summarize, the values to enter in the pressure far-field panel:

$P = 101325\ Pa$	$X_{component\_flow} = \cos\alpha$
$T = 288.15\ K$	$Y_{component\_flow} = \sin\alpha$
$M = 0.131927$	

Finally, we obtain Re_{num} after initialising the flow and checking the reference values given after computing from the pressure far-field boundary conditions. We are satisfied by the result since $\mathrm{Re}_{num} = 3072362$, which represents a relative error of only 0.9% compared to the experimental conditions.

FLUENT PARAMETERS

Model:

- Solver: *Coupled, implicit, Node-Based*
- Energy: *activated* (obligatory with pressure far-field boundary conditions)
- Viscous: *k-ε realizable* or *k-ω SST*

Material:

- Air: ρ → Ideal gas (forced by the pressure far-field BC) & $\mu = 1.7894*10^{-5}$ kg/m.s

Operating conditions:

- Operating pressure: 101325 Pa

Pressure far-field BC:

- Gauge Pressure: 0 Pa
- Mach number : 0.131927
- Temperature: 288.15 K
- X component of the flow direction: cos (α)
- Y component of the flow direction: sin (α)

Solve: → Control solution

- Flow: 2nd order upwind
- Turbulent kinetic energy: 1st order upwind
- Turbulent dissipation rate: 1st order upwind

b) Boundary layer

At the very beginning, we undertook a coarse calculation at a neutral angle of attack. The aim was not only to get a broad view of the problem but mainly to get an order of magnitude of the *Skin Friction Coefficient.* Consequently, we deduce an estimated value for the wall shear stress. The idea is to finally determine what should roughly be the height of the first node close to the wall, in order to respect the meshing requirement of $y^+ \approx 1$. Wall shear stress and skin friction coefficients are related by:

$$\tau_w = \frac{1}{2}\rho C_f U_\infty^{\,2}$$

τ_w *Wall shear stress* (Pa)
C_f *Skin friction coefficient*
ρ *Fluid density*
U_∞ *Free – stream velocity* $(m.s^{-1})$

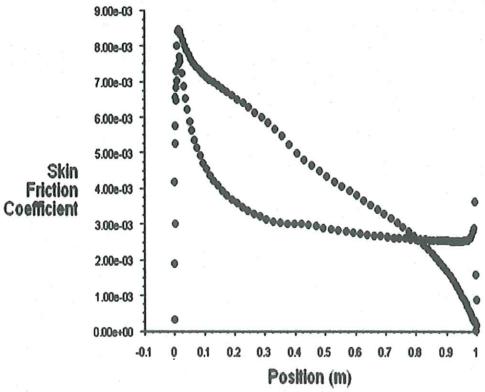
The velocity to be considered is the relative velocity between the airfoil and the free-stream passing around the body, and that is: $U_\infty = 44.87675\ m/s$. The following expression provides us with the relationship between the skin friction coefficient and the *friction velocity*. This last parameter is the missing link between the dimensionless wall distance y^+ and the distance to the wall from the adjacent cell, y. y^+ and y are related as follow:

$$u_* = \sqrt{\frac{\tau_w}{\rho}}$$
$$y^+ = \frac{u_*}{\nu} y$$

u_* *Friction velocity* $(m.s^{-1})$
ν *Local kinematic viscosity*
y^+ *Dimensionless Wall distance*
y *Distance to the nearest wall*

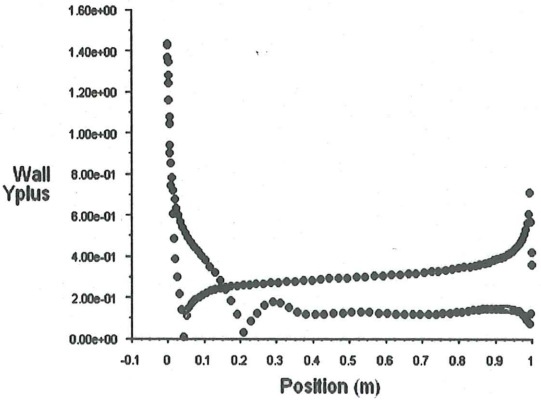
We have to bear in mind that the calculation is node-based and therefore the calculation is done in the middle of each cell so the height of the first cell is actually $2y$.

A value of $y^+ \approx 1$ is the uttermost to be considered, the first step is to recuperate the value of C_f from the coarse calculation. Its distribution along the profile's surface is represented by the plot on the right. Evidently, the front of the airfoil generates important friction since the flow is dramatically slowed down at the leading edge which explains the important peak. For our calculation, we take that highest value so as not to overestimate the first height:



$$\left.\begin{array}{l} C_f = 8.5\times10^{-3} \\ \tau_w = 10.47 \\ u_* = 2.92344 \end{array}\right\} \quad y^+ \approx 1 \ \Rightarrow \ y \approx 4.99\times10^{-6}$$

We rounded this value up to: $y = 5\times10^{-6}$ and imposed the distance to the first wall adjacent cell. The following plot represents the y^+ distribution along the airfoil's upper and lower surfaces for the highest incidence of 24°. Just as expected, the values of y^+ are in the correct interval since they are surrounding the desired value of 1. As a direct consequence of the high values of C_f in that region, the peak is maintained and y^+ reaches 1.43. However, all the values are contained such as $y^+ \in [0;1.43]$, which is more than acceptable since the first cell always remains in the viscous sublayer ($y^+ < 5$). To conclude, we can validate the choice of first height so that:



$$\boxed{y_{1st\_height} = 5\times10^{-6}}$$

c) Evolution of the lift coefficient

The final aim is to compare the pressure distributions obtained numerically to those measured during the experimental tests. However, we still need to determine which of the turbulence models is the most capable of capturing the different features of the flow. Moreover, the most challenging range for a model's reliability is at high angles of attack. The adverse pressure gradients become increasingly high and cause the flow to separate. A circular flow develops and the separation bubble is a very high turbulent region. In order to compare the k-ε realizable and k-ω SST model abilities, we have calculated the lift coefficients of each model for a series of angles of attack going from -12 to 24 degrees. The following figure also contains the experimental results obtained by Pinkerton, which are our reference in this study.

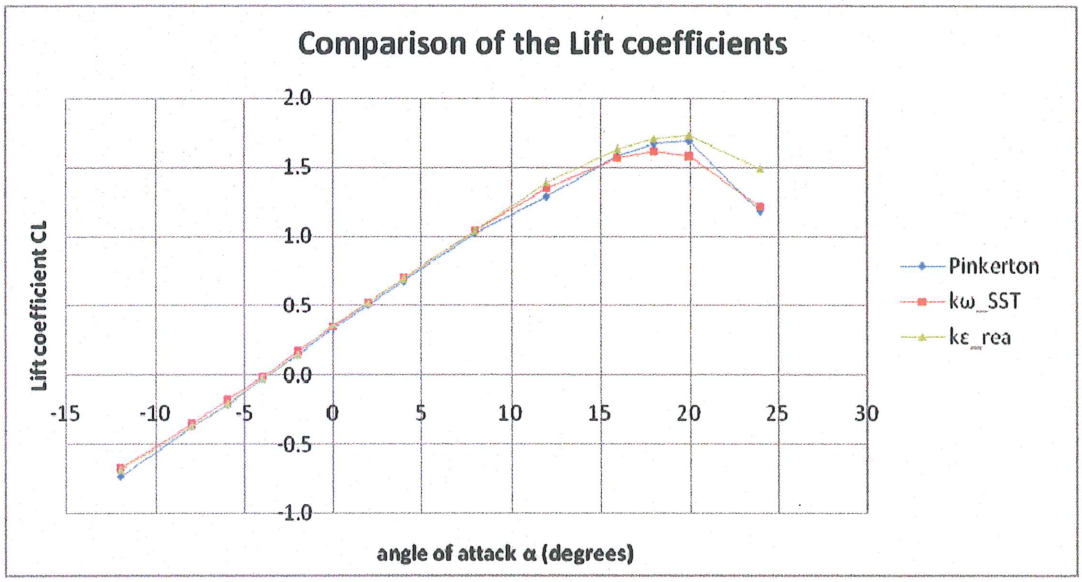


Figure 39: Numerical and experimental lift coefficients with respect to α

From the curves represented on *figure 39*, we can make several observations. First of all, the two models *kε-realizable* and *kω-SST* are very accurate for the prediction of C_L , as long as the flow remains attached. Indeed, that range of angles of attack is characterized by the linear portion of the curves and it approximately corresponds to $\alpha \in [-12;15]$ degrees. The two predictions are practically identical and none of the models can be favoured.

The second portion of the curve is deviating from the previous linear tendency, separation occurs at different incidences according to the turbulence model. At an incidence of about 12 degrees, a trailing edge stall occurs very early for the *kω-SST* model. Indeed, if we observe the evolution of the experimental C_L data, the curve remains linear well after 10°. At the contrary the *kω-SST* curve seems to foresee separation and slowly deviates. This gets noticeable on the C_L curve at around 15-18 degrees, but by analysing the data we have for smaller incidences, the velocity vectors indicate that the process has already started.

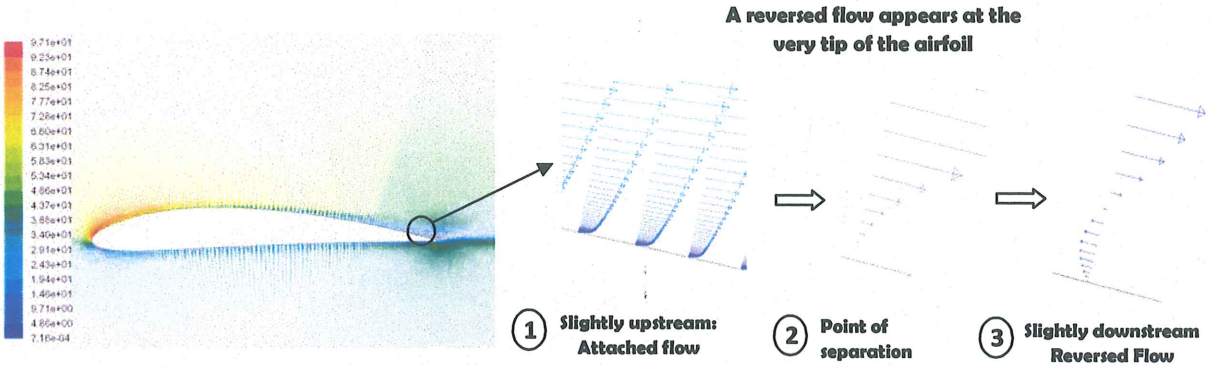


Figure 40: Velocity vectors for the kω SST model at α = 12°

As for the *kε-realizable* model, at 12 degrees it computes a fully attached flow with no sign of close separation. As we just mentioned, these characteristics of the flow may not have a direct impact on the C_L values at 12 degrees since they are very close, 1.35 and 1.39 for the *kω-SST* and *kε-realizable* respectfully. Nevertheless, we conclude that the *kω-SST* model anticipates separation which can be considered as a slight drawback of this particular turbulence model since as a consequence the C_L curve will be in advance in the separation region, which leads to a small underestimation of the values.

The incidence is gradually increasing and with it, the separation point moves upstream. At the incidence of 20 degrees, all three C_L curves are now declining, sign of a separated flow where the boundary layer has detached from the airfoil surface. Let us again compare the flow close to the airfoil surface for both models. As a relevant comparison, we draw a few path lines in that region to highlight the importance of the separation bubbles.

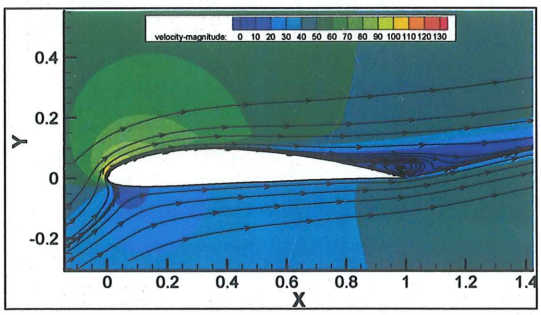


Figure 41: kε_realizable at α = 20 degrees

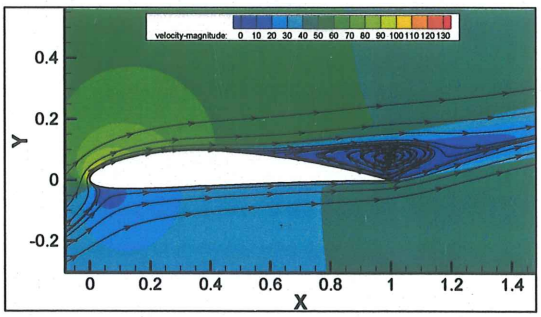


Figure 42: kω SST model at α = 20 degrees

From *figure 39 (C_L / α)*, up to now we have observed that the *kω SST* model was slightly anticipating separation at about 12 degrees. Thus, the maximum lift coefficient is not as high as it should be, $C_{L\max\_SST} \approx 18°$ instead of 20° but the results' accuracy remains very satisfying. Indeed, in the entire separation region ($18° \rightarrow 24°$) the SST curve follows the experimental one very well. At the contrary, the *kε_realizable* model which gave an almost exact value of $C_{L\max}$, once in the separation region the model is not capable of correctly evaluating the flow properties. *Figure 39 (C_L / α)* indicates that the lift coefficients are too high, from which we conclude that the model is

underestimating the importance of separation and simply not capturing the turbulent effects. This assumption is corroborated by the comparison of *figures 41 and 42*. The separation bubble of the *kε-realizable* model is much smaller than the *kω-SST* model. This last model provides accurate results since it is in close agreement with *R. M Pinkerton's* data. Consequently, it is chosen for the rest of the comparisons to the experimental information. We will now continue to evaluate the relevance of the shear stress transport model (SST) from the pressure distribution data.

d) Pressure distributions

Being able to obtain accurate numerical predictions of the pressure distributions along the surface of an airfoil is fundamental. It can be extremely useful for designing an airfoil for instance. Indeed, the boundary layer equations can be solved without actually knowing the airfoil shape provided the pressure distribution is given. As we know by now, a profile which results in early separation is clearly undesired due to the associated loss in lift and increase in drag. Consequently, the designer first prescribes the desired pressure distribution and then calculates the boundary layer to insure that there will be no separation. Indeed there are ways for him to check that no separation is present in the flow. When the boundary layer approaches the separation point, it develops the so called Goldstein singularity. In a very simplified way, it can be seen on the model of the boundary layer equation by the absence of the viscous term.

He will solve the so called–inverse problem, to obtain the shape of the airfoil from the prescribed pressure distribution. This inverse design principle is a widespread method but its reliability depends on numerical accuracy.

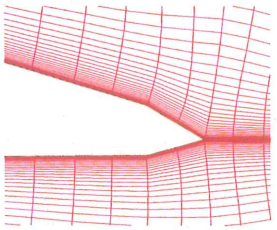
In the experimental report of *R. M Pinkerton*, data has been collected for many incidences. We have compared the *kω-SST* model's results to those distributions.

From the previous lift coefficient values we may guess that the distributions will be similar. Indeed, C_L is directly related to the pressure coefficients that vary all along the airfoil's surface by the following relation: $C_L = \int C_p n.k \, dA$ *'A'* represents the wing surface area, *'n'* the normal vector pointing into the wing and *'k'* the vertical unit vector, normal to the free-stream pressure.

This expression defines C_L as being the integral of vertical pressure forces over the entire wetted surface area of the wing. We deduce that the lift coefficient is directly related to the surface area enclosed by the C_P distribution. Therefore, since the C_L values are relatively close to the experimental data, we expect the same from the pressure distributions. We superposed the curves, which can be found just after the following observations we made about them.

Pressure distributions where: $\alpha \in [-12°;12°]$

We may recall that all of these incidence are contained within the linear part of the C_L versus α curve. The flow is fully attached and the values of C_L very close from the experimental data. All the pressure distributions are quasi identical in shape to the experimental ones. However, we notice that for every incidence, the domain is a bit shifted downwards due to an underestimation of all the pressure coefficients. This is the case for every curve. It can be attributed to the geometry of the NACA 4412 airfoil we created. In reality, the trailing edge of the airfoil has a certain thickness since an infinitely fine airfoil is impossible to manufacture. While generating the geometry, we had to close the domain somehow and a net ending of the profile, with a vertical line, seemed unwise. Therefore, the


Trailing edge

the trailing edge we created. This representation seems to generate slightly more circulation than desired since the curves, lower than expected, would correspond to a slightly higher incidence. Despite this slight error, we consider the results as relatively satisfying in this linear part of the incidence range. Yet there can be improves and in future studies it would be interesting to consider the effect of different trailing edge representations.

Pressure distributions at: $\alpha = 16°, 18°, 20°$

These three plots are contained in the separation region, after the onset of separation at $C_{L\max}$. The pressure coefficients distributions are very good since almost equal to the test data. The previous tendency of underestimating the C_P values is damped. This observation supports our assumption that the trailing edge was modifying the amount of circulation. Indeed, once the flow separates all the portion of the airfoil downstream of the separation point hardly contributes to the production of lift. Consequently, the geometrical approximation we made at the trailing edge has little effect. From 16 to 20 degrees, the distribution gets closer and closer to the experimental one. To conclude, the kω-SST model has proved to be very reliable in this critical range of incidences.

Pressure distribution at: $\alpha = 24°$

The incidence of 24° is the highest we simulated and corresponds to a highly turbulent flow. The flow separates very early from the airfoil's surface, at about 25% of the chord. *Figure 43* illustrates the velocity field and several stream-traces that highlight the presence of an important separation bubble.

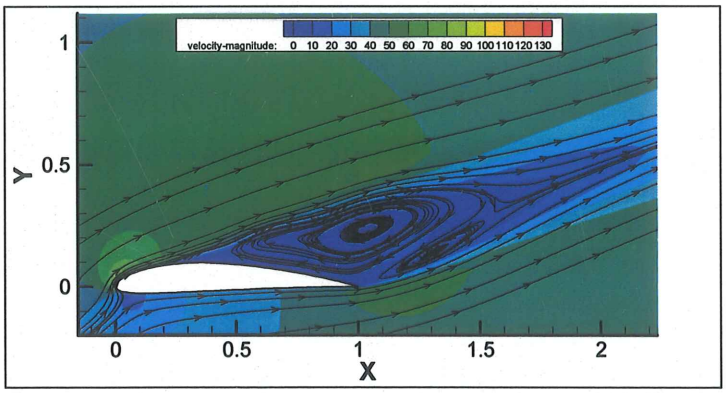


Figure 43: kω-SST model at α = 24 degrees

The pressure coefficients are perfectly matching the data for the lower surface but the results for the upper surface are much less good than previously. This is particularly the case for $C_P \in [0, 0.3]$ of the chord; probably due to the difference in the numerically predicted separation bubble and the real one. They have different impacts on the upstream flow.

To conclude we may say that the *kω-SST* turbulence model was capable of correctly estimating the amount of separation for most of the range of high incidences that follow the onset of separation. Despite the 24 degrees distribution of pressure coefficients, the inverse design procedure could realistically rely on numerical prediction obtained thanks to the *kω-SST* model. However, the trailing edge effect caused by the thickness representation would need to be corrected. In addition, the incidence from which the model is not reliable any more would need to be determined.

Pressure distributions along the NACA 4412 airfoil

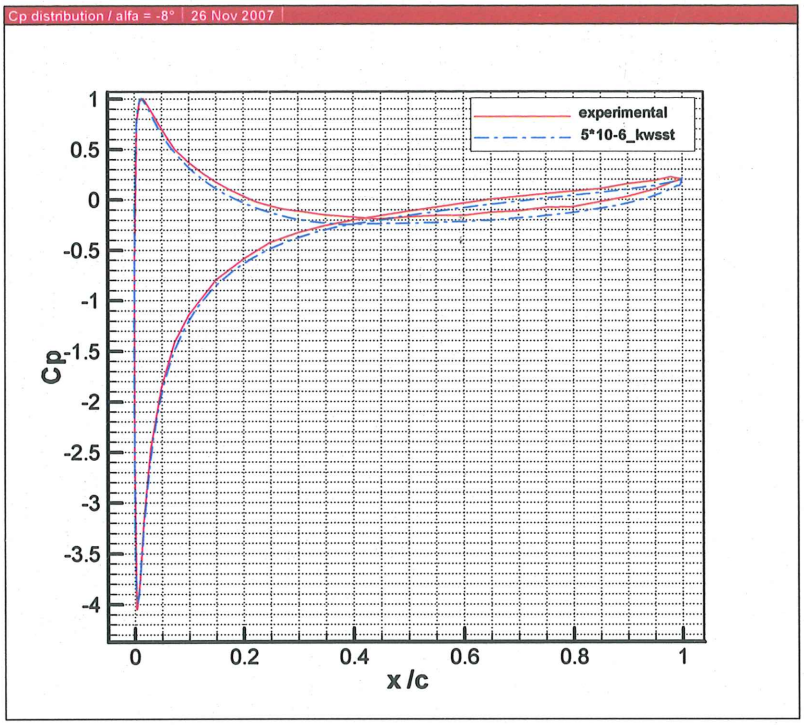


Figure 44: Pressure distributions at - 8 degrees

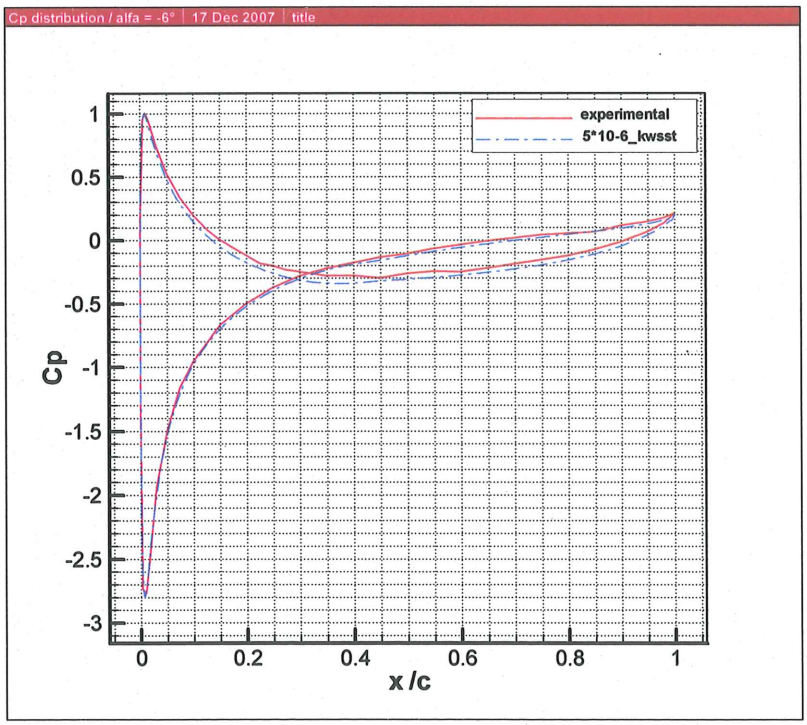


Figure 45: Pressure distributions at - 6 degrees

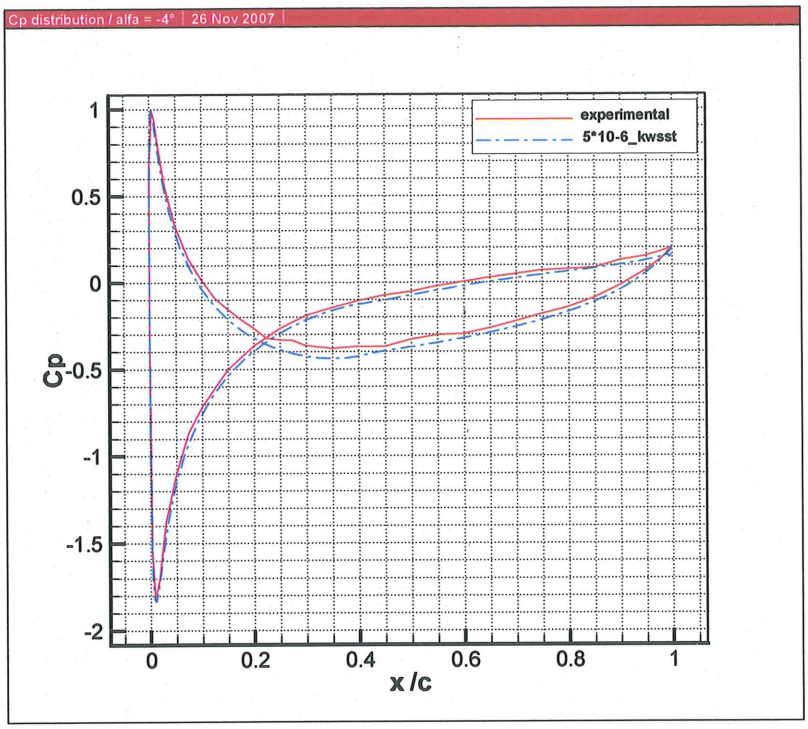


Figure 46: Pressure distributions at - 4 degrees

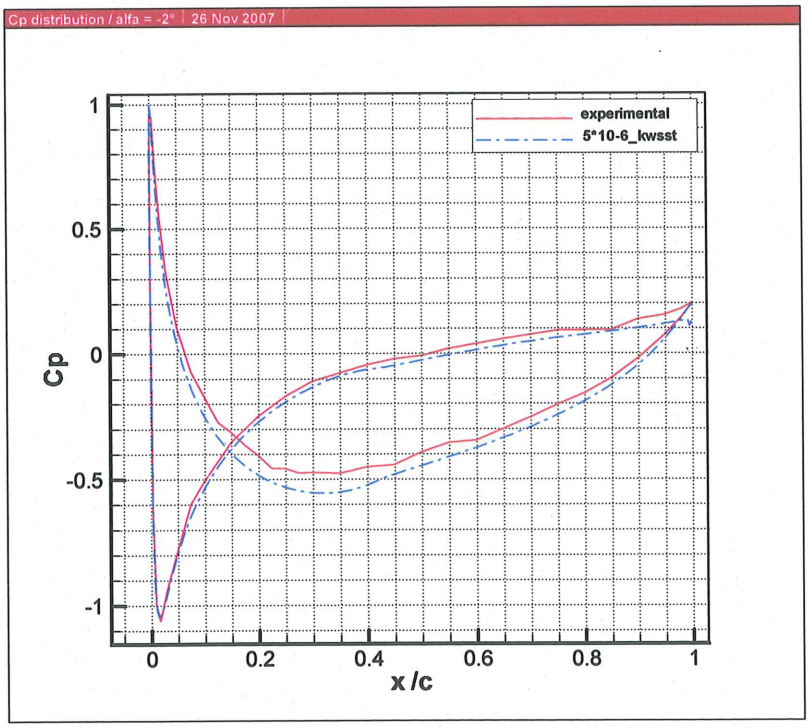


Figure 47: Pressure distributions at - 2 degrees

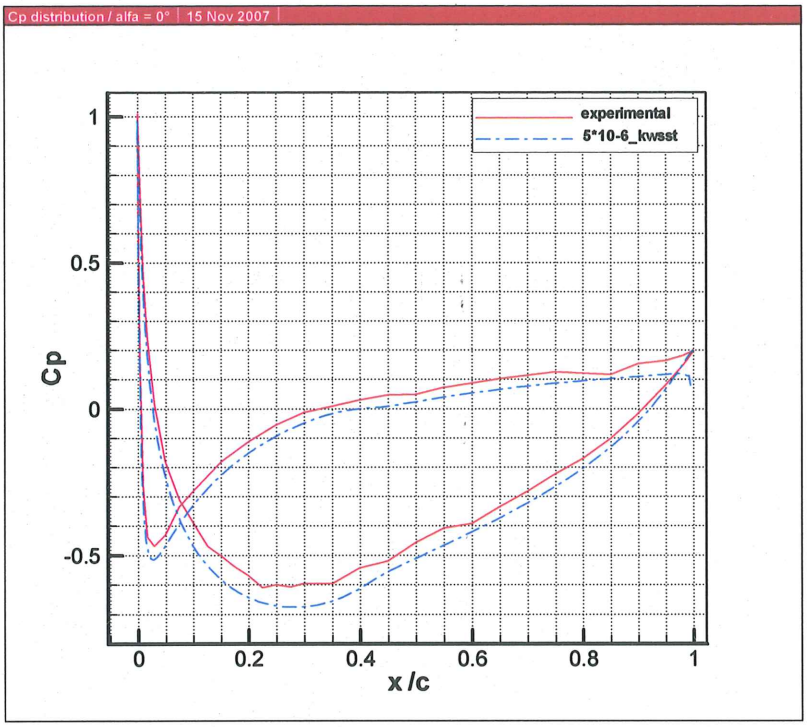


Figure 48: Pressure distributions at 0 degrees

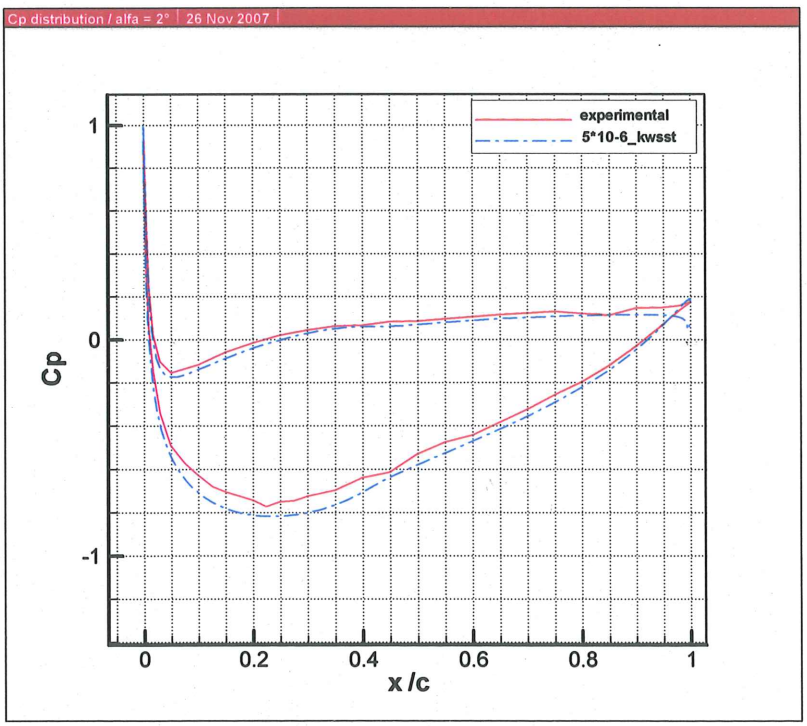


Figure 49: Pressure distributions at 2 degrees

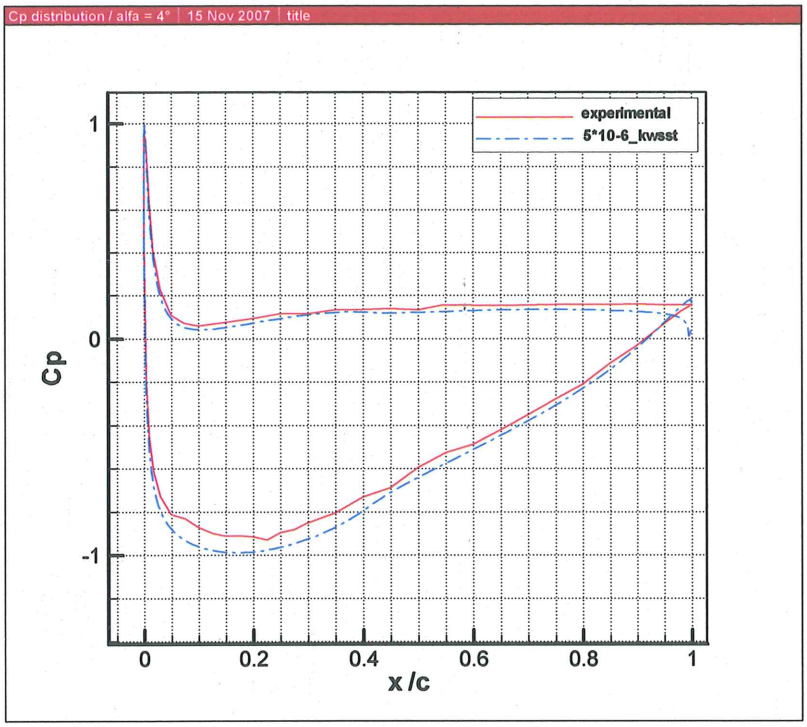


Figure 50: Pressure distributions at 4 degrees

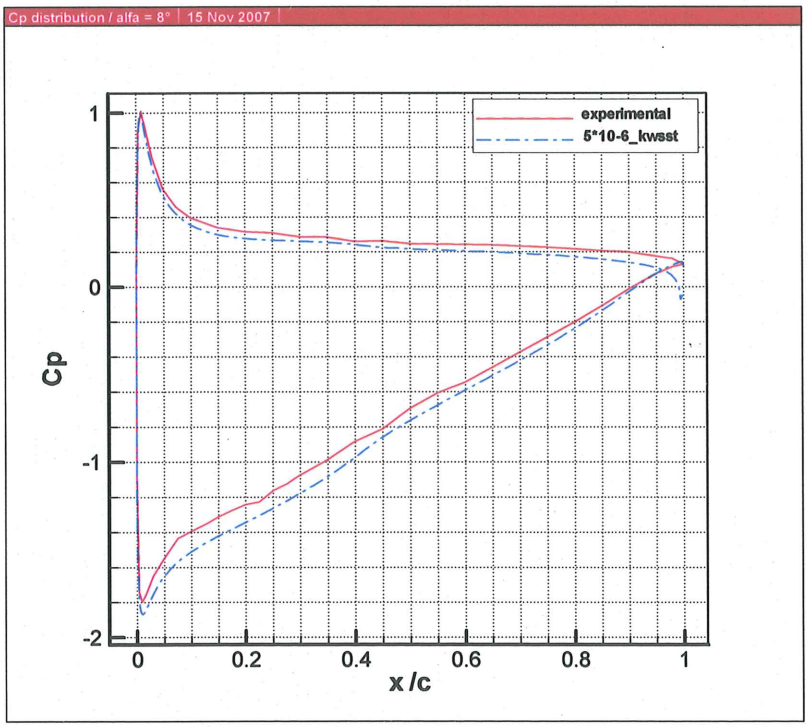


Figure 51: Pressure distributions at 8 degrees

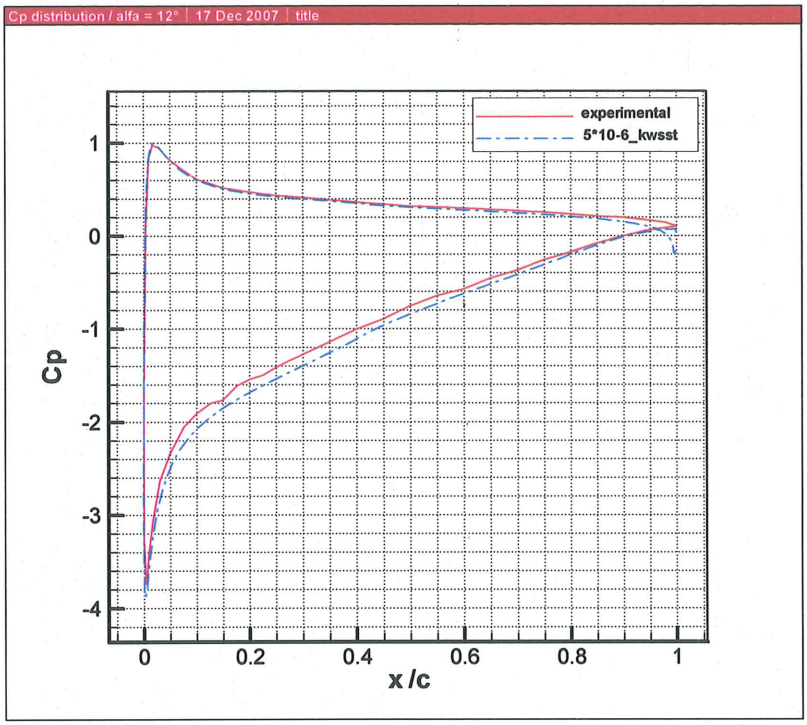


Figure 52: Pressure distributions at 12 degrees

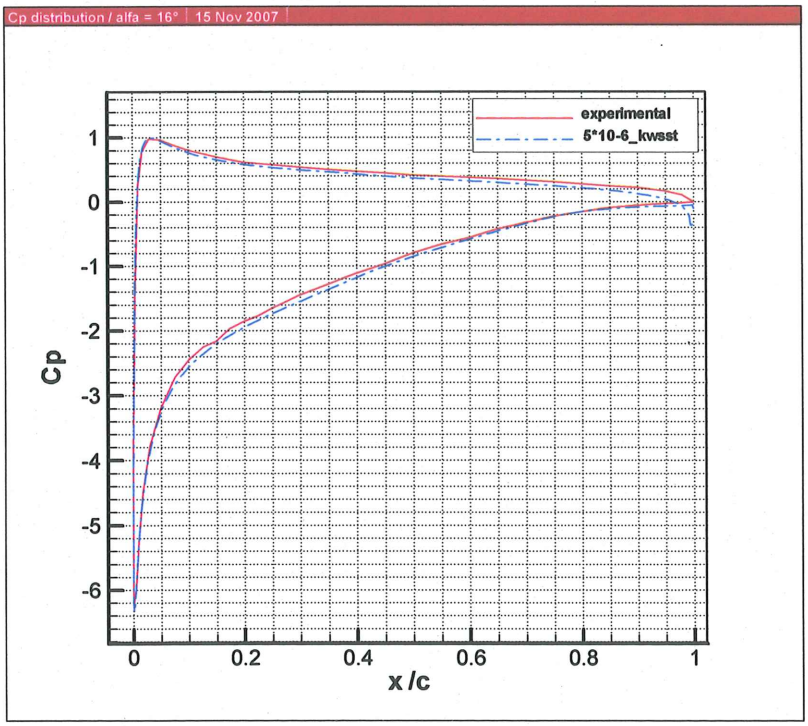


Figure 53: Pressure distributions at 16 degrees

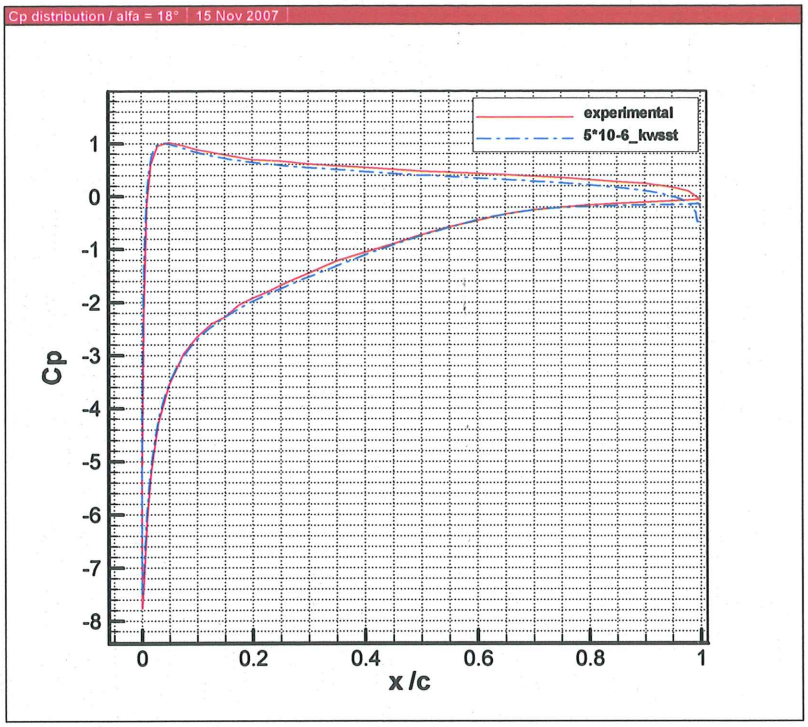


Figure 54: Pressure distributions at 18 degrees

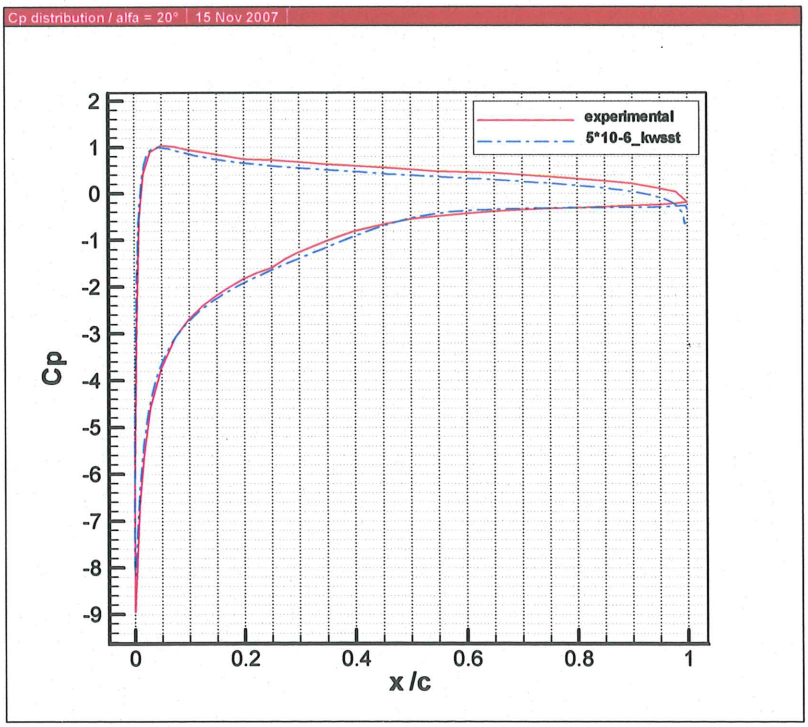


Figure 55: Pressure distributions at 20 degrees

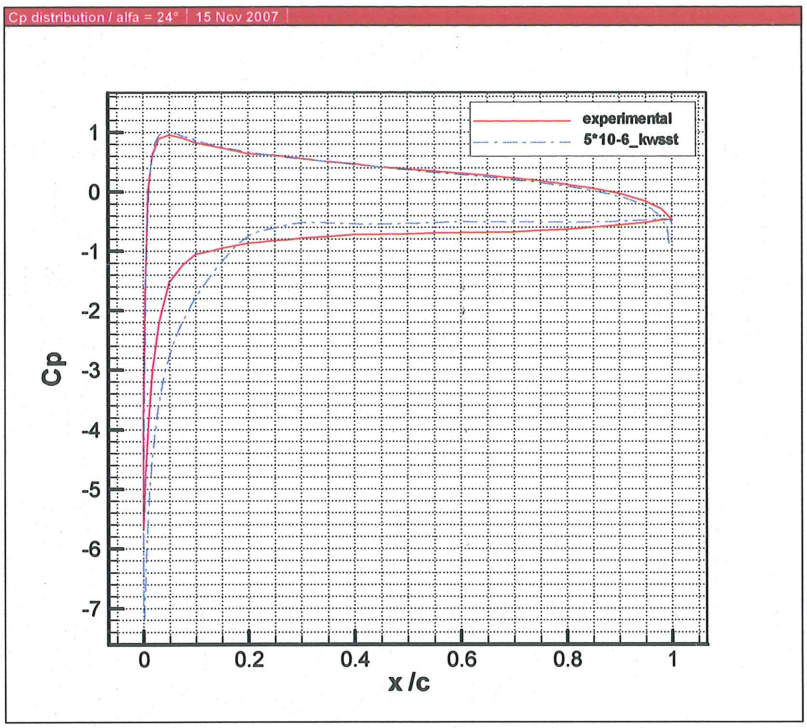


Figure 56: Pressure distributions at 24 degrees

CONCLUSION

The inviscid study resulted in the creation of a program, based on the Karman Trefftz theory that can be useful while making inviscid computations. Indeed, the CFD domain's boundaries introduce an error on the numerical results. Therefore, the user is required to enter the relative error that he considers as acceptable, which differs from one application to the other. As a result, the program provides him with the corresponding distances according to which the CFD domain is defined. The numerical simulations we conducted corroborated the program's predictions. We fixed Cp's relative error to 5% so that the domain has a negligible influence. Then the simulation parameters such as the discretization schema and the order of accuracy were modified in order to correspond to an airfoil application. Finally, the pressure distributions along the surface were found to be very similar to the theoretical expectations. We may deduce that the Karman Trefftz transformation is a valuable theory since it can help us to define the CFD domain of any inviscid calculation around an airfoil. It is also a very good reference to compare the exactitude of the numerical results.

From the viscous results we obtained, we may now draw conclusions relying on the data we gathered and analysed throughout the numerical simulations. We had to make a choice as for which turbulence model to use, based on the comparison between the numerical and experimental lift curves. To conclude, the kω-SST model is evidently the best model for the prediction of the flow across a whole range of incidences. Indeed, at high angles of attack, the amount of separation is close to the experiments. As for the k-ε realizable model, it is used in most of the turbulent flow calculations due to its robustness, economy and reasonable accuracy. However, its performance has appeared to be very poor for solving the flow around an airfoil at high incidence. Indeed, for non-equilibrium boundary layers, it will tend to predict the onset of separation too late although not so badly in our study, but above all it under-predicts the amount of separation. Separation is a critical component of turbulent aerodynamic flows and cannot be neglected. It influences the overall performance of many devices, such as aerodynamic bodies like wings for instance, but also diffusers, turbine blades...etc. The aerodynamic performance is very affected but separation also has a strong influence on multi phase phenomena or wall heat transfer for example. The kε-realizable model predicts reduced separation that leads to an optimistic prediction of machine performance. The consequences can be dangerous in many applications. If the prediction of wing stall on an airplane is not only inexact but higher than in reality, it may be catastrophic.

REFERENCES

PINKERTON Robert M, 'Calculated and measured pressure distributions over the midspan section of the NACA 4412 airfoil', naca-report-563, 1937

MUNK, MILLER Max M, ELTON W, 'The variable density wind tunnel of the National Advisory Committee for Aeronautics', naca-report-227, 1926

ABBOTT, IRA H, 'Airship model tests in the variable density wind tunnel', naca-report-394, 1932

YUKSELEN M. Adil, ERIM M.Z., 'A general iterative method to design Karman-Trefftz and Joukowski airfoils', Int.J. for numerical methods in Engineering, Vol.20, s.1349-1368, 1984

YON S., KATZ J., PLOTKIN A., 'Effect of Airfoil (Trailing-Edge) Thickness on the Numerical Solution of Panel Methods Based on the Dirichlet Boundary Condition', AIAA Journal, Vol. 30 No. 3 PP. 697-702, 1992

SHIH, LIOU, SHABBIR, YANG, 'A new Eddy-Viscosity Model for High Reynolds Number Turbulent Flows – Model Development and Validation', Computers Fluids, pp 227-238, 1995

APPENDIX A

1. Karman-Trefftz transformation: Matlab programs

Main.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MAIN PROGRAM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

Chord = 1;

%%%%%%%%%%%%%%%%%%%%%%% 1/ DEDUCING LIFT FROM INCIDENCE %%%%%%%%%%%%%%%%%%%

%% INPUTS: Airfoil geometric and flow (incidence) characteristics
[F,G,m,name,N1,AlfaD] = textread('airfoil_data.txt','%7f %7f %7f %11c %*s %f %*s %f',1)

%% Compute the Karman-Trefftz transformation function with complex
[z,U_z,x1_zeta,x_zeta,y_zeta,U_zeta,zeta,beta,b,XLE,Cu,Cp] = ktreff(AlfaD,F,G,m,Chord,N1)

%%%%%%%%%%%%%%%%%%%%%% 2/ DEDUCING INCIDENCE FROM LIFT %%%%%%%%%%%%%%%%%%%%

% %% INPUTS: Airfoil geometric characteristics and lift coefficient
% [F,G,m,name,N1,CL] = textread('airfoil_data.txt','%7f %7f %7f %11c %*s %f %*s %f',1)
%
% %% Calculation of the flow incidence from the lift coefficient
% [AlfaD,Gamma] = incidence(CL,F,G,m,Chord)
%
% %% Compute the Karman-Trefftz transformation function with complex
% [z,U_z,x1_zeta,x_zeta,y_zeta,U_zeta,zeta,beta,b,XLE,Cu,Cp] = ktreff(AlfaD,F,G,m,Chord,N1)

%%%%%%%%%%%%%%%% 3/ COMMON PART: CFD DOMAIN DIMENSIONS %%%%%%%%%%%%%%%%%%%%
%% Calculation of the coefficients
[Gamma,CL] = coeff(AlfaD,F,G,m,Chord,beta,b,XLE)

%% Domain calculation
[IX,JY,f,Cp_max] = domain (XLE,m,b,AlfaD,F,G,beta,N1,Gamma)

%%%%%%%%%%%%%%%%%%%%%%%%%%%% 4/ USEFULL FIGURES %%%%%%%%%%%%%%%%%%%%%%%%%%%

% %% CIRCLE REPRESENTATION
% hold on
% grid on
% figure(1)
% title('cylinder to transform')
% plot(real(z),imag(z),'b')
% %,real(z_line),imag(z_line),'r')
%
% %% AIRFOIL REPRESENTATION
% daspect([1 2 1])
% hold on
% grid on
% figure(2)
% title('Karman-Trefftz airfoil')
% Xlabel('x')
% Ylabel('y')
% plot(x_zeta(2:(N1-1)),y_zeta(2:(N1-1)),'r')
% hold off

% %% PRESSURE DISTRIBUTION ALONG THE AIRFOIL (Cu = 1 - Cp)
% hold on
% daspect([1 5 1])
% grid on
% figure(3)
% title('Pressure distribution')
% Xlabel('X / c')
```

```
% Ylabel('Cu = 1 - Cp')
% plot(x_zeta(2:(N1-1)),Cu(2:(N1-1)),'r');
% hold off
%
% %% VELOCITY DISTRIBUTION ALONG THE AIRFOIL
% hold on
% daspect([1 1 1])
% grid on
% figure(4)
% title('Velocity distribution')
% Xlabel('X / c')
% Ylabel('U')
% plot(x_zeta(2:(N1-1)),U_zeta(2:(N1-1)),'r');
% hold off

%%%%% 5/ TEXT FILE OUTPUT: Coordinates and velocities in both planes  %%%%%
%%%%%%%%%%%%%%%%%%%% and aerodynamic coefficients %%%%%%%%%%%%%%%%%%%%%%%%%

%% AIRFOIL LIFT AND PITCHING MOMENT COEFFICIENT
% fid = fopen('E:\sophia\ITU\Analytic airfoils\Matlab\result1.txt','w');
% fprintf(fid,'KARMAN-TREFFTZ TRANSFORMATION\n\n\n')
% fprintf(fid,'F = %6.6f\n',F)
% fprintf(fid,'G = %6.6f\n',G)
% fprintf(fid,'m = %6.6f\n',m)
% fprintf(fid,'alfa = %6.6f\n\n',AlfaD)
%
% %fprintf(fid,'Airfoil Coordinates and velocities\n\n\n')
% fprintf(fid,'   Xc          Yc          Uc          x_zeta       y_zeta       Cu\n\n')
%
% VECT = [real(z(1:(N1-1)));imag(z(1:(N1-1)));U_z(1:(N1-1));x1_zeta(1:(N1-1));y_zeta(1:(N1-
1));Cu(1:(N1-1))];
% VECT_last  = [real(z(N1));imag(z(N1));U_z(N1);x1_zeta(N1);y_zeta(N1);U_zeta(N1)];
%
% fprintf(fid,'%6.6f    %6.6f    %6.6f    %6.6f    %6.6f    %6.6f\n',VECT);
% fprintf(fid,'%6.6f    %6.6f    %6.6f    %6.6f    %6.6f    %6.6f\n\n\n',VECT_last);
% fprintf(fid,'Airfoil coefficients\n\n')
% fprintf(fid,'Circulation:                          Gamma = %6.5f\n',Gamma)
% fprintf(fid,'Lift coefficient:                     CL = %6.5f\n',CL)
% fclose(fid)

%%%%%%%%%%%%% 6/ CREATION OF THE JOURNAL TO AUTOMATICALLY %%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%% GENERATE THE AIRFOIL GEOMETRTY %%%%%%%%%%%%%%%%%%%%%%%
%% GAMBIT journal file containing the airfoil coordinates (500 points)

fid = fopen('E:\sophia\ITU\Analytic airfoils\Matlab\coordinates.txt','w')

fprintf(fid, '/ Journal file for GAMBIT\n')
fprintf(fid, '/ Identifier KT_00_15_40\n\n\n')

zcoord=0*ones(1,500);
VECT = [x_zeta;y_zeta;zcoord];
fprintf(fid,'vertex create coordinates      %6.6f      %6.6f        %6.6f\n',VECT);
fclose(fid)

%%%%%%%% 7/ EXPORTATION OF THE PRESSURE COEFFICIENTS AND AIRFOIL  %%%%%%%%%
%%%%%%%%%%%%%%%% COORDINATES FOR COMPARISONWITH FLUENT RESULTS %%%%%%%%%%%%
%% Cp distribution along the airfoil

fid = fopen('E:\sophia\ITU\Analytic airfoils\Fluent\airfoil\KT-00-15-
40\bon_500points\analytical data\alfa_10.dat','w')

CPvect = [x_zeta;Cp];
fprintf(fid,' %6.6f     %6.6f\n',CPvect);
fclose(fid)
```

Ktreff.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%%%%%%%%%%%%%%%% AIRFOIL COORDINATES AND PRESSURE COEFFICIENTS  via VELOCITIES
%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

function[z,U_z,x1_zeta,x_zeta,y_zeta,U_zeta,zeta,beta,b,XLE,Cu,Cp] =
ktreff(AlfaD,F,G,m,Chord,N1)

beta = atan(G / (1+F));
b = Chord / (2 * m) * (1 - (F / (1 + F)) ^ m);
f = F / b;
g = G / b;
XLE = m * b - Chord;
AlfaR = AlfaD * pi / 180.0;
n = N1 - 1;

% first define i=sqrt(-1) and parameters a(circle radius)
i = complex(0.0,1.0);
a = b *(1.0 + (f / b)) / cos(beta);
V_inf = 1
NN = n / 2 + 1;       % Aim: have the exact same number of points on the upper and lower
surfaces
if (F == 0)
    NN = n / 2;
end

dTeta = 2 * pi / n;             % Angle increment

for i = 1 : NN                  % First loop calculating the values for each point

    theta = -(i - 1) * dTeta;   % Indirect trigonometric direction distribution of teta

    for k = 1:2         % Second loop first considers the lower surface and then the upper
points

        %% CALCULATION OF AIRFOIL COORDINATES
        SinT = sin(theta);     % SinT, CosT and ARG are terms used to simplify the expression
of R
        CosT = cos(theta);
        ARG = -F * CosT + G * SinT;
        R = ARG + sqrt(1 + 2 * F + ARG *ARG);

        ii = i;
        if (k == 2)
            ii = n - i + 2;
        end
        z(ii) = b * complex(R * CosT,R * SinT);
        zp = (z(ii) + b).^ m;
        zm = (z(ii) - b).^m;
        zeta = m * b * (zp + zm)/(zp - zm);        % Karman-Trefftz transformation

        x1_zeta(ii) = real(zeta);                  % Airfoil coordinates
        x_zeta(ii) = real(zeta) + XLE + 1;
        y_zeta(ii) = imag(zeta);

%% CALCULATION OF AIRFOIL VELOCITIES
        dzetadz = 4 * m ^ 2 * b ^ 2 * ((z(ii) ^ 2 - b ^ 2) ^ (m - 1)) / ((z(ii) + b) ^ m -
(z(ii) - b) ^ m) ^ 2;
        Mod_dzetadz = sqrt(real(dzetadz) ^ 2 + imag(dzetadz) ^ 2);        % Module of dzetadz
        theta1 = theta - AlfaR - asin(sqrt((F ^ 2 + G ^ 2)/((1 + F) ^ 2 + G ^ 2)) * sin(theta
+ atan(G / F)));
        U_z(ii) = 2 * (sin(theta1) + sin(AlfaR + beta));     % Velocity in the z plane
        U_zeta(ii) = U_z(ii) / Mod_dzetadz;          % Velocities on the airfoil in the zeta
plane

       % Pressure coefficient
        Cu(ii) = U_zeta(ii) * U_zeta(ii);
        Cp(ii) = 1 - Cu(ii);
        theta = - theta;
    end
end
```

incidence.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AIRFOIL INCIDENCE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

function [AlfaD,Gamma] = incidence(CL,F,G,m,Chord)

% Usefull parameters
b = Chord / (2 * m) * (1 - (F / (1 + F)) ^ m);
A2 = (1 + F) ^ 2 + G ^ 2;           % Circle radius squared defined as a 'module'
A = sqrt(A2);                       % Circle radius
beta = atan(G / (1+F));

% Calculation of the airfoil incidence
Gamma = (CL * Chord) / 2;
AlfaR = asin(Gamma / (4*pi*A*b)) - beta;
AlfaD = AlfaR * (180 / pi);         % Airfoil incidence in degrees
```

Coeff.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%% AIRFOIL LIFT COEFFICIENT %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function [Gamma,CL] = coeff(AlfaD,F,G,m,Chord,beta,b,XLE)

% Usefull parameters
Cbb = Chord / b;
AlfaR = AlfaD * (pi / 180);         % Airfoil incidence in radians
A2 = (1 + F) ^ 2 + G ^ 2;           % Circle radius squared defined as a 'module'
A = sqrt(A2);                       % Circle radius

% Lift coefficient
Gamma = 4 * pi * A * b * sin(AlfaR + beta); % Circulation so that the Kutta condition is
satisfied (ref Jouk_p4)
CL = 2 * Gamma / Chord;                     % Lift coefficient

% % Pitching moment
% C1 = 4 * (m ^ 2 - 1) / 3 / Cbb ^ 2 * pi * sin(2 * AlfaR); % C1 and C2 are terms used to
simplify the expression of CM0
% C2 = F * cos(AlfaR) - G * sin(AlfaR);
% C2 = 8 * pi * A / (Cbb ^ 2) * C2 * sin(AlfaR + beta);
% CM0 = C1 + C2;                                            % Pitching moment coefficient
% CM_LE = CM0 - CY * (abs(XLE) / Chord);                    % Pitching moment coefficient at
the leading edge (ref KT_p9)
% CM_cb4 = CM0 - CY * (abs(XLE) / Chord - 0.25);            % Pitching moment coefficient at
the quarter chord
% Alfa0 = -beta * (180 / pi);
```

domain.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% COMPUTATIONAL DOMAIN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

%%% This function is used to determine the domain's dimensions necessary for a CFD
calculation. %%% The allowable pressure coefficient error is fixed. The maximum value of Cp
```

```
along one of the 4 %%% domain lines is considered. As long as Cp_max is too small the domain
gets smaller and     %%% smaller, until the error becomes acceptable (economically).

function[IX,JY,f,Cp_max] = domain (XLE,m,b,AlfaD,F,G,beta,N1,Gamma)

AlfaR = AlfaD * pi / 180.0;

f = F / b;
g = G / b;
ci = complex(0,1);
V_inf = 1;
A2 = (1 + F) ^ 2 + G ^ 2;
A = sqrt(A2);
mb = m*b;
m1 = 1 / m;

IX = -80;
JY = 62;
NT = - 2 * IX + 1;

quest = 0;

while (quest ==0)

    for i = 1:NT

        % Inverse coordinates of the domain limit (line), in the z plane
        X = i + IX - 1;
        Y = JY;
        Zeta(i) = complex(X,Y)

        z1 = Zeta(i) + mb;
        z2 = Zeta(i) - mb;
        zz = (z1 / z2) ^ m1;

        zD(i) = b * (1 + zz) / (1 - zz);

        % Velocities along the limit

        z1 = zD(i) + b;
        z2 = zD(i) - b;
        zz = z1 ^ m - z2 ^ m;
        dZetadz(i) = 4 * (mb) ^ 2 * ((z1 * z2) ^ (m-1)) / (zz) ^ 2;
        Mod_dZetadz(i) = abs(dZetadz(i));

        % Velocities in the z plane
        zD1(i) = (zD(i) + F*b - ci * G*b) * exp(-ci * AlfaR);

        w_zD1(i) = V_inf * (1 - A^2 * b^2/ zD1(i)^2) + (ci * Gamma) / (2 * pi * zD1(i));
        VzD(i) = abs(w_zD1(i));
        VZeta(i) = VzD(i) / Mod_dZetadz(i);

        % Pressure coefficient
        Cu(i) = VZeta(i) ^ 2;
        Cp_line(i) = 1 - Cu(i)
    end

    % Calculation of the maximum error on the line
    Cp_max = max(abs(Cp_line(:)));


if (abs(Cp_max) < 0.0047)      % The result is too accurate and can be more economic
        JY = JY - 0.1;
    else
        if (abs(Cp_max) > 0.0053)  % The error is too large compared to its permitted value
            JY = JY + 0.1;
            quest = 1;
        else
            quest = 1;
        end
    end
end
```

2. Numerical results of Karman-Trefftz transformation (*matlab*)

	γ = 0% & δ = [5%,10%,15%,20%]								γ = 5% & δ = [5%,10%,15%,20%]							
	KT 00 05 40		KT 00 10 40		KT 00 15 40		KT 00 20 40		KT 05 05 40		KT 05 10 40		KT 05 15 40		KT 05 20 40	
CL	α°	X_up	α°	X_up	α°	X_up	α°	X_up	α°	X_up	α°	X_up	α°	X_up	α°	X_up
0	0	1.5	0	2.2	0	2.8	0	3.3	-5.8167	2	-5.9064	2.5	-5.9815	2.9	-6.0402	3.4
0.1	0.87875	1.7	0.84656	2.3	0.81552	2.8	0.78571	3.3	-4.9422	2.6	-5.0639	2.9	-5.1698	3.2	-5.258	3.6
0.2	1.7577	3.2	1.6933	3.3	1.6312	3.5	1.5716	3.8	-4.0675	3.7	-4.2212	3.8	-4.3578	4	-4.4757	4.2
0.3	2.6371	5	2.5404	5	2.4472	5	2.3577	5.1	-3.1925	5	-3.3781	5.1	-3.5456	5.2	-3.6931	5.3
0.4	3.5171	5.1	3.3881	6.8	3.2637	6.9	3.1443	6.9	-2.3168	6.6	-2.5344	6.6	-2.7328	6.6	-2.9101	6.7
0.5	4.3979	8.8	4.2365	8.8	4.0809	8.7	3.9315	8.7	-1.4403	8.3	-1.69	8.2	-1.9194	8.2	-2.1265	8.2
0.6	5.2798	10.8	5.0858	10.7	4.8989	10.7	4.7194	10.7	-0.56271	10	-0.84475	10	-1.1052	9.9	-1.3421	9.9
0.7	6.1629	12.8	5.9363	12.7	5.7179	12.7	5.5083	12.6	0.31607	11.8	0.0016529	11.8	-0.29002	11.7	-0.55689	11.7
0.8	7.0475	14.9	6.7881	14.8	6.5381	14.7	6.2982	14.7	1.1963	13.7	0.84935	13.6	0.52635	13.6	0.22938	13.5
0.9	7.9338	17	7.6414	16.9	7.3597	16.8	7.0892	16.8	2.0782	15.6	1.6985	15.5	1.344	15.5	1.0168	15.4
1	8.822	19.2	8.4964	19.1	8.1827	19	7.8817	18.9	2.962	17.6	2.5494	17.5	2.1632	17.4	1.8056	17.3
1.1	9.7123	21.4	9.3533	21.3	9.0075	21.2	8.6756	21.1	3.8479	19.7	3.4022	19.5	2.9841	19.4	2.5959	19.3
1.2	10.605	23.7	10.212	23.6	9.8341	23.4	9.4713	23.3	4.7361	21.8	4.257	21.6	3.8069	21.5	3.3879	21.3
1.3	11.5	26	11.074	25.9	10.663	25.7	10.269	25.6	5.6269	23.9	5.1141	23.7	4.6316	23.6	4.1817	23.4
1.4	12.399	28.4	11.938	28.2	11.494	28.1	11.068	27.9	6.5205	26.2	5.9738	25.9	5.4587	25.7	4.9775	25.5
1.5	13.3	30.9	12.804	30.7	12.327	30.4	11.87	30.2	7.4172	28.4	6.8361	28.2	6.2881	27.9	5.7754	27.7
1.6	14.205	33.4	13.674	33.1	13.163	32.9	12.674	32.7	8.3172	30.7	7.7014	30.5	7.1202	30.2	6.5757	29.9
1.7	15.113	35.9	14.547	35.6	14.002	35.4	13.481	35.1	9.2208	33.1	8.5699	32.8	7.955	32.5	7.3784	32.2
1.8	16.025	38.5	15.423	38.2	14.844	37.9	14.29	37.6	10.128	35.6	9.4418	35.2	8.7929	34.9	8.1839	34.5
1.9	16.941	41.2	16.303	40.8	15.69	40.5	15.102	40.2	11.04	38	10.317	37.6	9.6341	37.3	8.9922	36.9
2	17.862	43.9	17.187	43.5	16.539	43.1	15.918	42.8	11.956	40.6	11.197	40.1	10.479	39.7	9.8037	39.3

	γ = 10% & δ = [5%,10%,15%,20%]								γ = 15% & δ = [5%,10%,15%,20%]							
	KT 10 05 40		KT 10 10 40		KT 10 15 40		KT 10 20 40		KT 15 05 40		KT 15 10 40		KT 15 15 40		KT 15 20 40	
CL	α°	X_up	α°	X_up	α°	X_up	α°	X_up	α°	X_up	α°	X_up	α°	X_up	α°	X_up
0	-11.515	2.6	-11.688	2.9	-11.836	3.3	-11.947	3.7	-16.987	3.2	-17.234	3.4	-17.441	3.7	-17.614	4
0.1	-10.653	3.2	-10.857	3.4	-11.035	3.7	-11.175	4	-16.144	3.5	-16.421	3.8	-16.657	4	-16.859	4.3
0.2	-9.791	4	-10.026	4.1	-10.234	4.3	-10.403	4.5	-15.302	4.2	-15.609	4.3	-15.874	4.5	-16.103	4.7
0.3	-8.9284	5	-9.1951	5.1	-9.4332	5.2	-9.631	5.3	-14.459	5	-14.796	5.1	-15.09	5.2	-15.347	5.4
0.4	-8.0652	6.3	-8.3632	6.3	-8.6315	6.4	-8.8583	6.5	-13.616	6.1	-13.983	6.1	-14.306	6.2	-14.591	6.3
0.5	-7.2013	7.7	-7.5307	7.7	-7.8292	7.7	-8.0851	7.8	-12.772	7.3	-13.169	7.3	-13.521	7.3	-13.834	7.3
0.6	-6.3363	9.3	-6.6973	9.2	-7.0261	9.2	-7.3112	9.2	-11.927	8.6	-12.355	8.5	-12.736	8.5	-13.076	8.5
0.7	-5.4702	10.9	-5.8629	10.8	-6.2221	10.8	-6.5364	10.7	-11.08	10	-11.539	9.9	-11.95	9.9	-12.318	9.8
0.8	-4.6027	12.5	-5.0272	12.5	-5.4169	12.4	-5.7607	12.3	-10.233	11.4	-10.722	11.3	-11.162	11.3	-11.558	11.2
0.9	-3.7336	14.3	-4.1901	14.2	-4.6104	14.1	-4.9837	14	-9.384	13	-9.9039	12.9	-10.373	12.8	-10.798	12.7
1	-2.8627	16.1	-3.3513	15.9	-3.8026	15.8	-4.2056	15.7	-8.5334	14.6	-9.0843	14.4	-9.5835	14.3	-10.037	14.2
1.1	-1.9897	17.9	-2.5108	17.8	-2.9931	17.6	-3.4259	17.5	-7.6809	16.2	-8.2629	16.1	-8.7921	15.9	-9.2736	15.8
1.2	-1.1145	19.8	-1.6682	19.7	-2.1818	19.5	-2.6447	19.3	-6.8263	18	-7.4397	17.8	-7.9989	17.6	-8.5092	17.4
1.3	-0.23691	21.8	-0.8235	21.6	-1.3686	21.4	-1.8617	21.2	-5.9695	19.8	-6.6145	19.5	-7.204	19.3	-7.7432	19.1
1.4	0.64341	23.8	0.023628	23.6	-0.55318	23.4	-1.0768	23.2	-5.1101	21.6	-5.787	21.3	-6.407	21.1	-6.9754	20.9
1.5	1.5267	25.9	0.87336	25.6	0.26452	25.4	-0.2899	25.1	-4.248	23.5	-4.9571	23.2	-5.6079	22.9	-6.2056	22.7
1.6	2.4131	28.1	1.7259	27.8	1.0847	27.5	0.49929	27.2	-3.3829	25.4	-4.1245	25.1	-4.8064	24.8	-5.4338	24.5
1.7	3.3029	30.3	2.5815	29.9	1.9076	29.6	1.2909	29.3	-2.5147	27.4	-3.2891	27.1	-4.0024	26.7	-4.6597	26.4
1.8	4.1964	32.5	3.4404	32.1	2.7334	31.7	2.0851	31.4	-1.6431	29.5	-2.4507	29.1	-3.1957	28.7	-3.8831	28.3
1.9	5.0938	34.8	4.3027	34.4	3.5624	34	2.882	33.6	-0.76783	31.6	-1.6091	31.2	-2.3862	30.7	-3.104	30.3
2	5.9955	37.2	5.1688	36.7	4.3946	36.2	3.6819	35.8	0.11136	33.8	-0.76397	33.3	-1.5735	32.8	-2.3222	32.4

	γ = 0% & δ = [5%,10%,15%,20%]								δ = 5% & γ = [0% 5%,10%,15%]							
	KT 00 05 40		KT 00 10 40		KT 00 15 40		KT 00 20 40		KT 00 05 40		KT 05 05 40		KT 10 05 40		KT 15 05 40	
CL	α°	X_down	α°	X_down	α°	X_down	α°	X_down	α°	X_down	α°	X_down	α°	X_down	α°	X_downn
0	0	1.6	0	2.3	0	2.9	0	3.4	0	1.6	-5.8167	2.3	-11.515	2.8	-16.987	3.4
0.1	0.87875	2.1	0.84656	2.6	0.81552	3.1	0.78571	3.5	0.87875	2.1	-4.9422	1.7	-10.653	2.6	-16.144	3
0.2	1.7577	3.5	1.6933	3.6	1.6312	3.9	1.5716	4.2	1.7577	3.5	-4.0675	2.8	-9.791	2.2	-15.302	3
0.3	2.6371	5.1	2.5404	5.1	2.4472	5.2	2.3577	5.3	2.6371	5.1	-3.1925	5	-8.9284	4.8	-14.459	4.4
0.4	3.5171	6.5	3.3881	6.6	3.2637	6.6	3.1443	6.7	3.5171	6.5	-2.3168	6.8	-8.0652	7	-13.616	7.1
0.5	4.3979	8	4.2365	8	4.0809	8.1	3.9315	8.1	4.3979	8	-1.4403	8.5	-7.2013	9	-12.772	9.5
0.6	5.2798	9.4	5.0858	9.5	4.8989	9.5	4.7194	9.5	5.2798	9.4	-0.56271	10.2	-6.3363	10.9	-11.927	11.6
0.7	6.1629	10.8	5.9363	10.8	5.7179	10.9	5.5083	10.9	6.1629	10.8	0.31607	11.7	-5.4702	12.7	-11.08	13.6
0.8	7.0475	12.1	6.7881	12.1	6.5381	12.2	6.2982	12.3	7.0475	12.1	1.1963	13.2	-4.6027	14.4	-10.233	15.5
0.9	7.9338	13.3	7.6414	13.4	7.3597	13.5	7.0892	13.5	7.9338	13.3	2.0782	14.7	-3.7336	16	-9.384	17.3
1	8.822	14.5	8.4964	14.6	8.1827	14.7	7.8817	14.8	8.822	14.5	2.962	16.1	-2.8627	17.6	-8.5334	19.1
1.1	9.7123	15.6	9.3533	15.8	9.0075	15.9	8.6756	16	9.7123	15.6	3.8479	17.4	-1.9897	19.1	-7.6809	20.8
1.2	10.605	16.7	10.212	16.9	9.8341	17	9.4713	17.1	10.605	16.7	4.7361	18.7	-1.1145	20.6	-6.8263	22.5
1.3	11.5	17.8	11.074	17.9	10.663	18.1	10.269	18.3	11.5	17.8	5.6269	19.9	-0.23691	22	-5.9695	24.1
1.4	12.399	18.8	11.938	19	11.494	19.1	11.068	19.3	12.399	18.8	6.5205	21	0.64341	23.3	-5.1101	25.6
1.5	13.3	19.7	12.804	19.9	12.327	20.1	11.87	20.3	13.3	19.7	7.4172	22.2	1.5267	24.6	-4.248	27.1
1.6	14.205	20.6	13.674	20.8	13.163	21.1	12.674	21.3	14.205	20.6	8.3172	23.2	2.4131	25.9	-3.3829	28.5
1.7	15.113	21.4	14.547	21.7	14.002	22	13.481	22.2	15.113	21.4	9.2208	24.2	3.3029	27.1	-2.5147	29.9
1.8	16.025	22.2	15.423	22.5	14.844	22.8	14.29	23.1	16.025	22.2	10.128	25.2	4.1964	28.2	-1.6431	31.2
1.9	16.941	22.9	16.303	23.3	15.69	23.6	15.102	23.9	16.941	22.9	11.04	26	5.0938	29.3	-0.76783	32.5
2	17.862	23.6	17.187	24	16.539	24.4	15.918	24.7	17.862	23.6	11.956	26.9	5.9955	30.3	0.11136	33.7

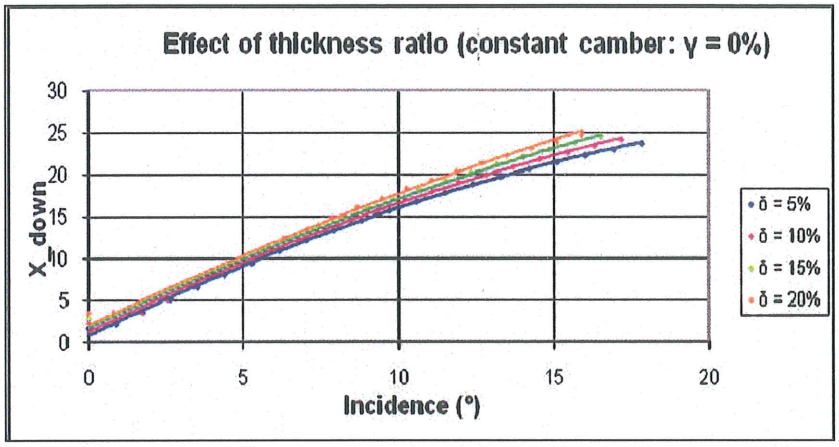
	γ = 0% & δ = [5%,10%,15%,20%]								δ = 5% & γ = [0% 5%,10%,15%]							
	KT 00 05 40		KT 00 10 40		KT 00 15 40		KT 00 20 40		KT 00 05 40		KT 05 05 40		KT 10 05 40		KT 15 05 40	
CL	α°	Y_up	α°	Y_up	α°	Y_up	α°	Y_up	α°	Y_up	α°	Y_up	α°	Y_up	α°	Y_up
0	0	1.5	0	2.2	0	2.8	0	3.3	0	1.5	-5.8167	2.1	-11.515	2.7	-16.987	3.2
0.1	0.87875	2.4	0.84656	1.8	0.81552	1.8	0.78571	2	0.87875	2.4	-4.9422	3.2	-10.653	3.9	-16.144	4.5
0.2	1.7577	6.3	1.6933	5.8	1.6312	5.2	1.5716	4.3	1.7577	6.3	-4.0675	6.3	-9.791	6.5	-15.302	6.6
0.3	2.6371	9.8	2.5404	9.6	2.4472	9.2	2.3577	8.8	2.6371	9.8	-3.1925	9.8	-8.9284	9.7	-14.459	9.6
0.4	3.5171	13.3	3.3881	13.1	3.2637	12.8	3.1443	12.6	3.5171	13.3	-2.3168	13.2	-8.0652	13.1	-13.616	12.9
0.5	4.3979	16.7	4.2365	16.5	4.0809	16.3	3.9315	16.1	4.3979	16.7	-1.4403	16.7	-7.2013	16.5	-12.772	16.3
0.6	5.2798	20.1	5.0858	19.9	4.8989	19.8	4.7194	19.6	5.2798	20.1	-0.56271	20.1	-6.3363	20	-11.927	19.7
0.7	6.1629	23.5	5.9363	23.3	5.7179	23.2	5.5083	23.1	6.1629	23.5	0.31607	23.5	-5.4702	23.4	-11.08	23.1
0.8	7.0475	26.8	6.7881	26.7	6.5381	26.6	6.2982	26.5	7.0475	26.8	1.1963	26.9	-4.6027	26.8	-10.233	26.5
0.9	7.9338	30.2	7.6414	30.1	7.3597	30	7.0892	29.9	7.9338	30.2	2.0782	30.3	-3.7336	30.2	-9.384	29.9
1	8.822	33.5	8.4964	33.4	8.1827	33.4	7.8817	33.3	8.822	33.5	2.962	33.6	-2.8627	33.6	-8.5334	33.4
1.1	9.7123	36.8	9.3533	36.8	9.0075	36.7	8.6756	36.6	9.7123	36.8	3.8479	37	-1.9897	37	-7.6809	36.8
1.2	10.605	40.1	10.212	40.1	9.8341	40	9.4713	40	10.605	40.1	4.7361	40.4	-1.1145	40.4	-6.8263	40.2
1.3	11.5	43.4	11.074	43.4	10.663	43.4	10.269	43.3	11.5	43.4	5.6269	43.7	-0.23691	43.8	-5.9695	43.6
1.4	12.399	46.7	11.938	46.7	11.494	46.7	11.068	46.6	12.399	46.7	6.5205	47.1	0.64341	47.2	-5.1101	47
1.5	13.3	50	12.804	50	12.327	49.9	11.87	49.9	13.3	50	7.4172	50.4	1.5267	50.5	-4.248	50.4
1.6	14.205	53.2	13.674	53.2	13.163	53.2	12.674	53.2	14.205	53.2	8.3172	53.7	2.4131	53.9	-3.3829	53.8
1.7	15.113	56.4	14.547	56.4	14.002	56.5	13.481	56.5	15.113	56.4	9.2208	57	3.3029	57.3	-2.5147	57.3
1.8	16.025	59.6	15.423	59.7	14.844	59.7	14.29	59.7	16.025	59.6	10.128	60.3	4.1964	60.6	-1.6431	60.7
1.9	16.941	62.8	16.303	62.8	15.69	62.9	15.102	62.9	16.941	62.8	11.04	63.5	5.0938	64	-0.76783	64.1
2	17.862	65.9	17.187	66	16.539	66.1	15.918	66.1	17.862	65.9	11.956	66.8	5.9955	67.3	0.11136	67.4

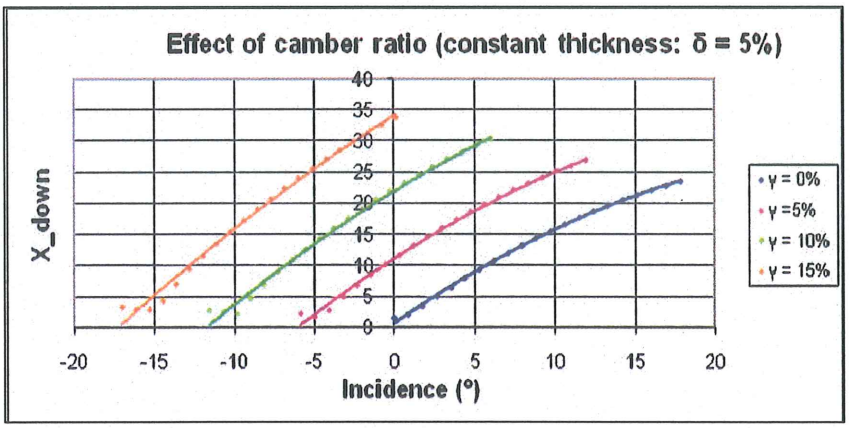
	γ = 0% & δ = [5%,10%,15%,20%]								δ = 5% & γ = [0% 5%,10%,15%]							
	KT 00 05 40		KT 00 10 40		KT 00 15 40		KT 00 20 40		KT 00 05 40		KT 05 05 40		KT 10 05 40		KT 15 05 40	
CL	α°	Y_low	α°	Y_low	α°	Y_low	α°	Y_low	α°	Y_low	α°	Y_low	α°	Y_low	α°	Y_low
0	0	1.5	0	2.2	0	2.8	0	3.3	0	1.5	-5.8167	2.2	-11.515	2.8	-16.987	3.4
0.1	0.87875	3.9	0.84656	4.4	0.81552	4.9	0.78571	5.4	0.87875	3.9	-4.9422	4.2	-10.653	4.7	-16.144	5.1
0.2	1.7577	7	1.6933	7.4	1.6312	7.7	1.5716	8.1	1.7577	7	-4.0675	7.2	-9.791	7.4	-15.302	7.6
0.3	2.6371	10.3	2.5404	10.6	2.4472	10.8	2.3577	11.1	2.6371	10.3	-3.1925	10.4	-8.9284	10.5	-14.459	10.6
0.4	3.5171	13.7	3.3881	13.9	3.2637	14.1	3.1443	14.3	3.5171	13.7	-2.3168	13.7	-8.0652	13.8	-13.616	13.8
0.5	4.3979	17	4.2365	17.2	4.0809	17.3	3.9315	17.5	4.3979	17	-1.4403	17.1	-7.2013	17.1	-12.772	17.1
0.6	5.2798	20.4	5.0858	20.5	4.8989	20.6	4.7194	20.8	5.2798	20.4	-0.56271	20.4	-6.3363	20.5	-11.927	20.4
0.7	6.1629	23.7	5.9363	23.8	5.7179	23.9	5.5083	24.1	6.1629	23.7	0.31607	23.8	-5.4702	23.8	-11.08	23.7
0.8	7.0475	27.1	6.7881	27.2	6.5381	27.3	6.2982	27.4	7.0475	27.1	1.1963	27.2	-4.6027	27.2	-10.233	27.1
0.9	7.9338	30.4	7.6414	30.5	7.3597	30.6	7.0892	30.7	7.9338	30.4	2.0782	30.6	-3.7336	30.6	-9.384	30.5
1	8.822	33.7	8.4964	33.8	8.1827	33.9	7.8817	34	8.822	33.7	2.962	33.9	-2.8627	34	-8.5334	33.9
1.1	9.7123	37	9.3533	37.1	9.0075	37.2	8.6756	37.3	9.7123	37	3.8479	37.3	-1.9897	37.4	-7.6809	37.3
1.2	10.605	40.3	10.212	40.4	9.8341	40.5	9.4713	40.6	10.605	40.3	4.7361	40.7	-1.1145	40.8	-6.8263	40.7
1.3	11.5	43.6	11.074	43.7	10.663	43.8	10.269	43.9	11.5	43.6	5.6269	44	-0.23691	44.2	-5.9695	44.1
1.4	12.399	46.9	11.938	47	11.494	47.1	11.068	47.2	12.399	46.9	6.5205	47.3	0.64341	47.5	-5.1101	47.5
1.5	13.3	50.2	12.804	50.3	12.327	50.4	11.87	50.5	13.3	50.2	7.4172	50.7	1.5267	50.9	-4.248	50.9
1.6	14.205	53.4	13.674	53.5	13.163	53.6	12.674	53.7	14.205	53.4	8.3172	54	2.4131	54.3	-3.3829	54.3
1.7	15.113	56.6	14.547	56.7	14.002	56.9	13.481	57	15.113	56.6	9.2208	57.3	3.3029	57.6	-2.5147	57.7
1.8	16.025	59.8	15.423	60	14.844	60.1	14.29	60.2	16.025	59.8	10.128	60.6	4.1964	61	-1.6431	61.1
1.9	16.941	63	16.303	63.1	15.69	63.3	15.102	63.4	16.941	63	11.04	63.8	5.0938	64.3	-0.76783	64.5
2	17.862	66.2	17.187	66.3	16.539	66.5	15.918	66.6	17.862	66.2	11.956	67.1	5.9955	67.7	0.11136	67.9

APPENDIX B

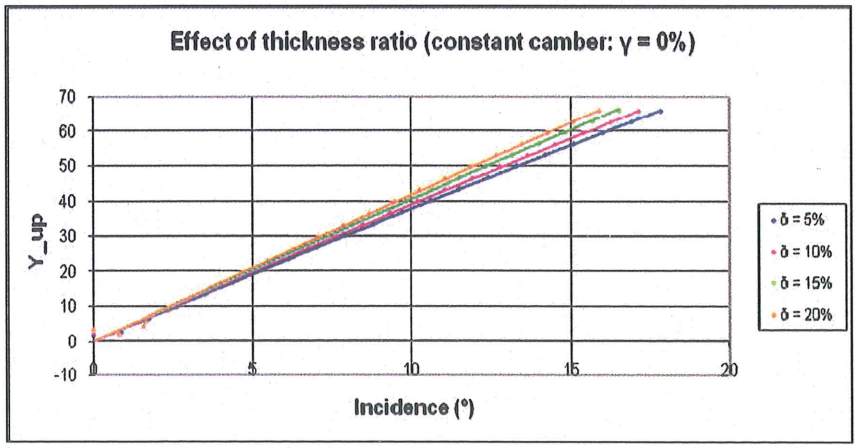
1. CFD domain dimensions as a function of incidence

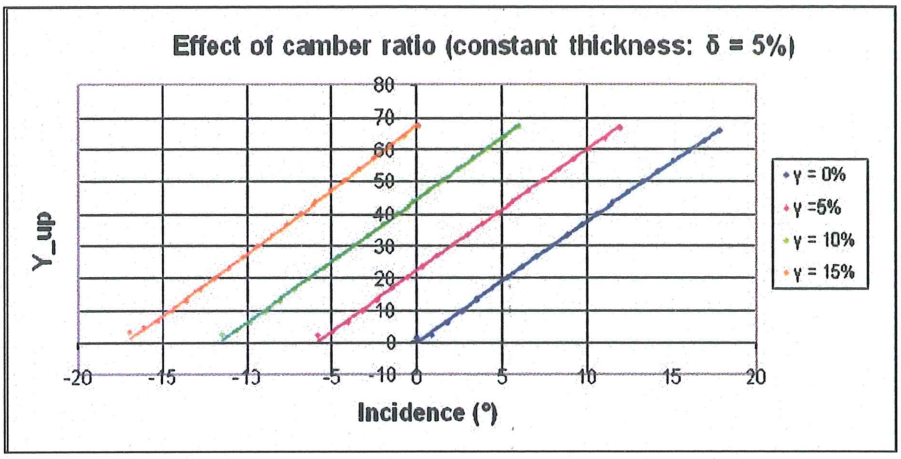
a) X_{down} / α



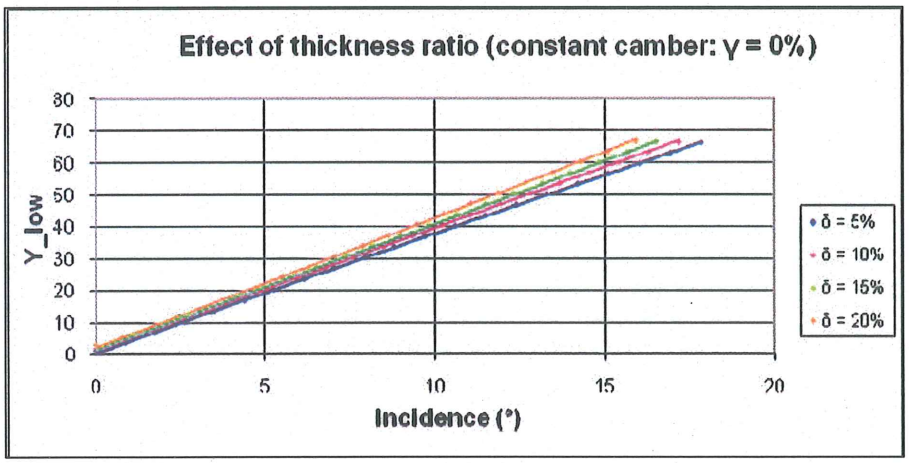


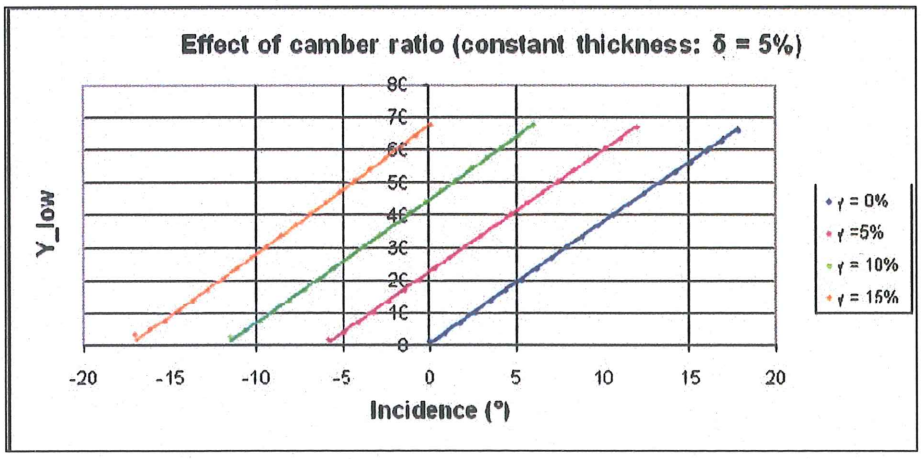
b) Y_{up} / α





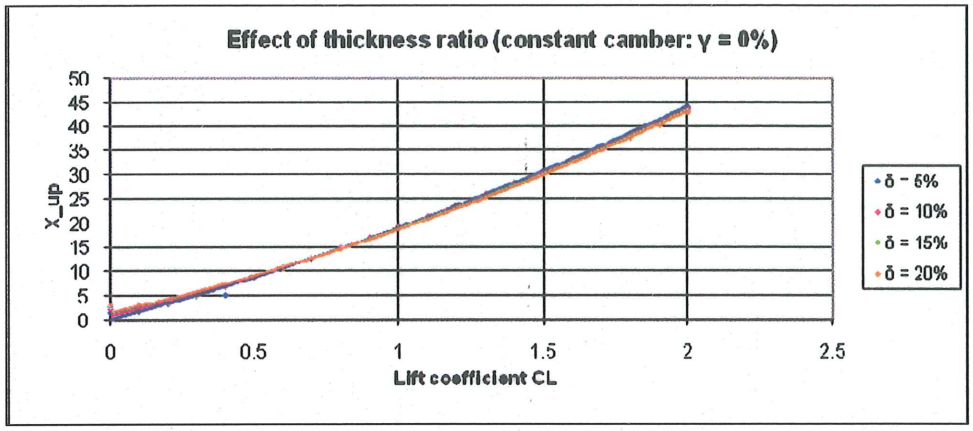
c) $\underline{Y_{down} / \alpha}$

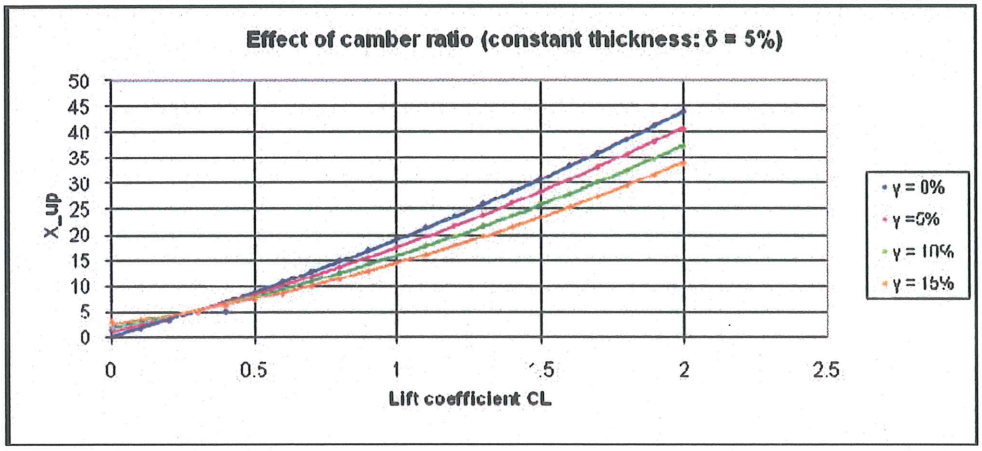




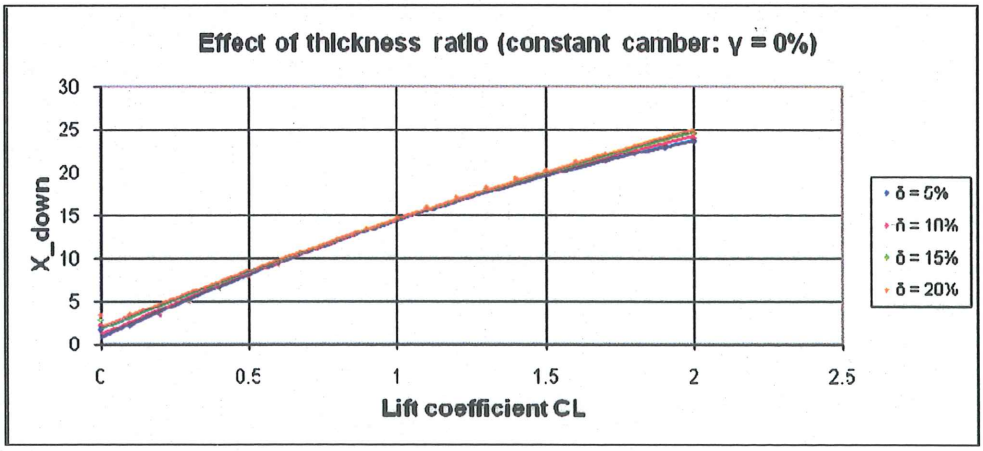
2. CFD domain dimensions as a function of the lift coefficient

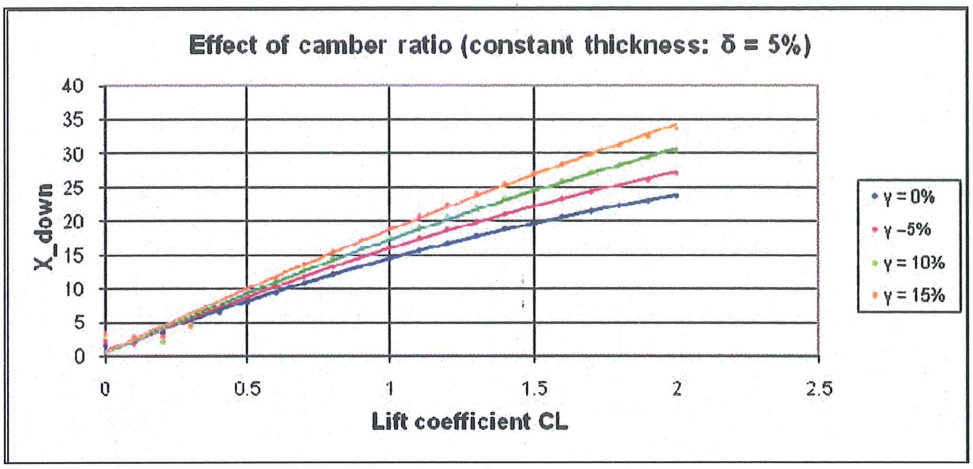
a) X_{up} / C_L



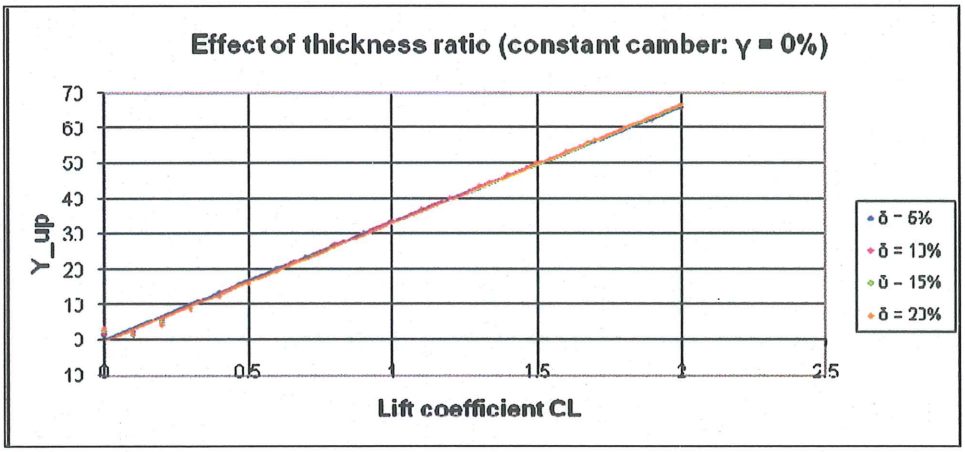


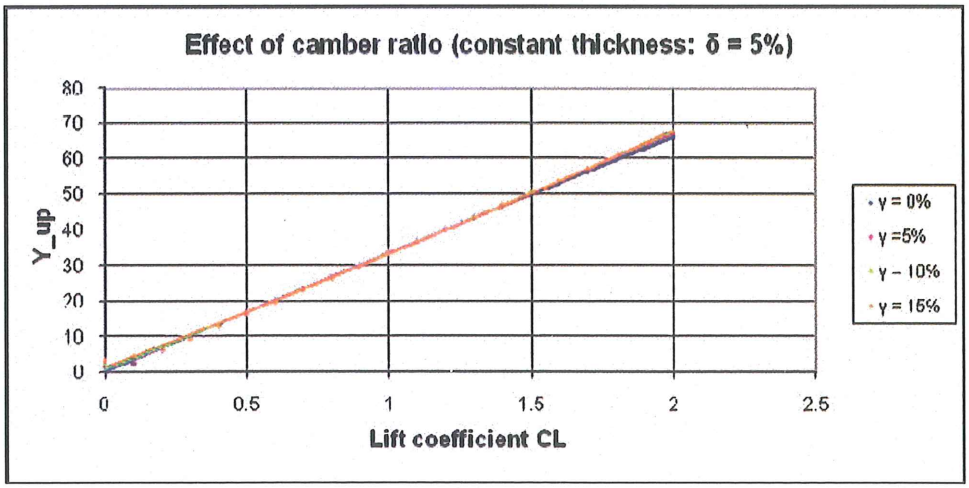
b) X_{down} / C_L



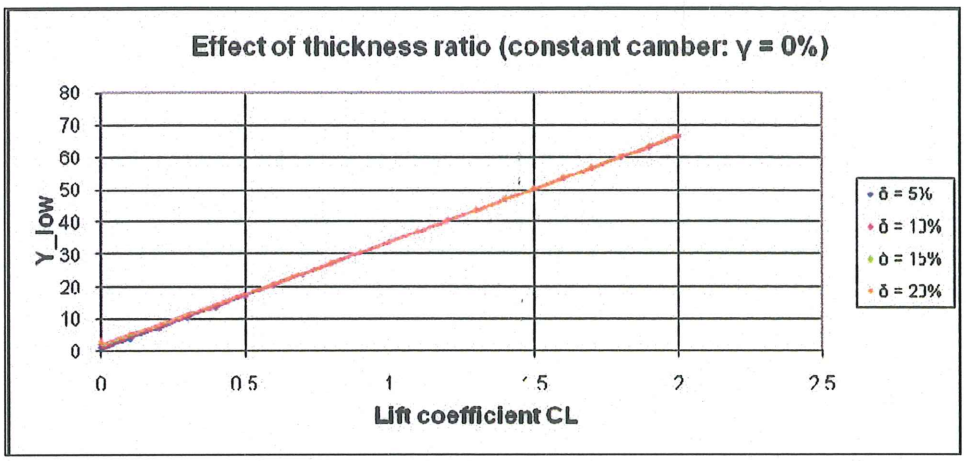


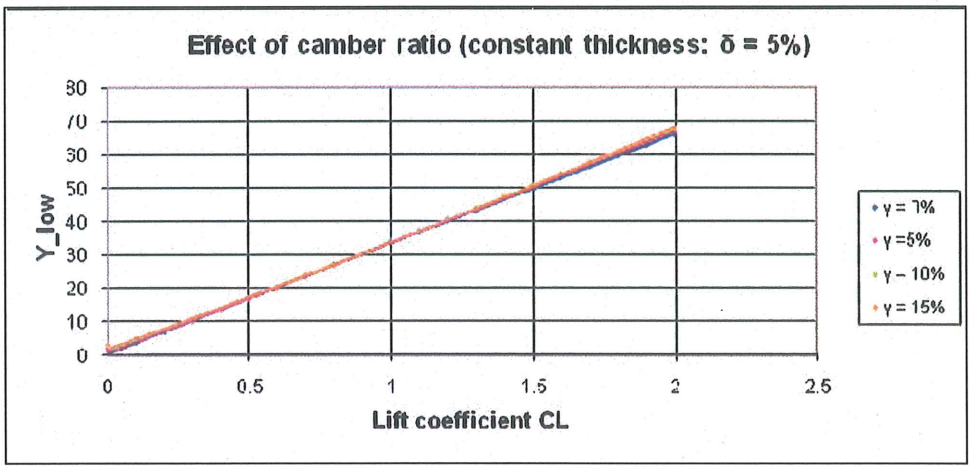
c) $\underline{Y_{up} / C_L}$





d) $\underline{Y_{down} / C_L}$





APPENDIX C

2D SPHERE

General specifications

- Material: air
- Viscous model: Inviscid
- Boundary conditions
  - Inlet [velocity inlet] : $V_\infty = 1$ m/s ($\alpha = 0°$)
  - Outlet [Pressure outlet] : $P_{out} = 1$ atm
  - Sphere ↔ wall
  - X axis ↔ Symmetry
- **Domain dimensions**

The pressure coefficient is one the main parameters we are interested in. In any case, the numerical study introduces a certain error which makes real conditions impossible to achieve. The simulation takes place in a limited area as opposed to infinite boundaries of reality, which impacts on the accuracy of the results. The pressure coefficient is expressed as follow:

$$Cp = 1 - \left(\frac{V}{V_\infty}\right)^2 \Rightarrow \lim_{x\to\infty} Cp \to 0$$

Indeed, infinitely far from the sphere, the velocity is equal to the free stream velocity. In the numerical case this is not achievable. However, a certain error on the pressure coefficient can be tolerated, for which the results accuracy remains acceptable. We will now undertake the analytical calculation that gives us the domain dimensions with respect to the error.

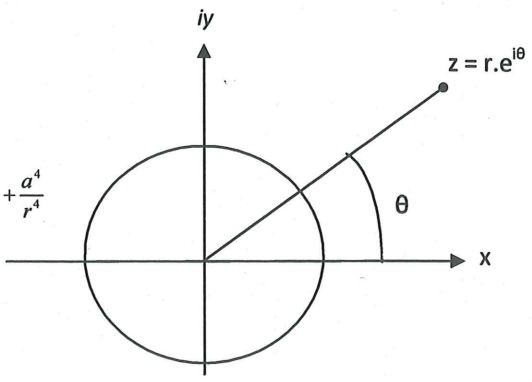
We consider the circle of radius *'a'* inscribed in a complex plane *(x,iy)*, where: $z = x + iy$.

$$w^*(z) = V_\infty.(1 - \frac{a^2}{z^2}) = V_\infty\left(1 - \frac{a^2}{r^2.e^{i2\theta}}\right)$$

$$\frac{w^*}{V_\infty} = 1 - \frac{a^2}{r^2} \times e^{-i2\theta} = 1 - \frac{a^2}{r^2}\cos(2\theta) + i\frac{a^2}{r^2}\sin(2\theta)$$

$$\left(\frac{w^*}{V_\infty}\right)^2 = \left(\frac{V}{V_\infty}\right)^2 = \left(1 - \frac{a^2}{r^2}\cos(2\theta)\right)^2 + \frac{a^4}{r^4}\sin^2(2\theta) = 1 - 2\frac{a^2}{r^2}\cos(2\theta) + \frac{a^4}{r^4}$$

$$Cp = 1 - \left(\frac{V}{V_\infty}\right)^2 \quad \Rightarrow \quad \boxed{Cp = 2\frac{a^2}{r^2}\cos(2\theta) - \frac{a^4}{r^4}}$$



We know that the maximum value of the pressure coefficient is located at the stagnation point and the minimum at the top of the circle:

$$Cp_{max} \rightarrow \max(\cos 2\theta) = +1 \quad \Rightarrow \quad 2\theta = 0 \rightarrow \theta = 0 \qquad Cp_{max} = 2\frac{a^2}{r^2} - \frac{a^4}{r^4} = \frac{2}{r^2} - \frac{1}{r^4}$$

$$Cp_{min} \rightarrow \min(\cos 2\theta) = -1 \quad \Rightarrow \quad 2\theta = \pi \rightarrow \theta = \pi/2 \qquad Cp_{min} = -2\frac{a^2}{r^2} - \frac{a^4}{r^4} = -\frac{2}{r^2} - \frac{1}{r^4}$$

By using these expressions, we deduce approximately what is the ratio of r (the domain's radius) over a (the circle's radius), for a given Cp error. The following results have been determined for characteristic errors.

r	Cpmin	Cpmax
10	-0.02	+0.02
20	-0.005	+0.005

In our study an error of 0.005 on the pressure coefficient is satisfying. The results remain sufficiently accurate. Consequently, we must impose: $\boxed{r/a = 20}$

1) Discretization scheme

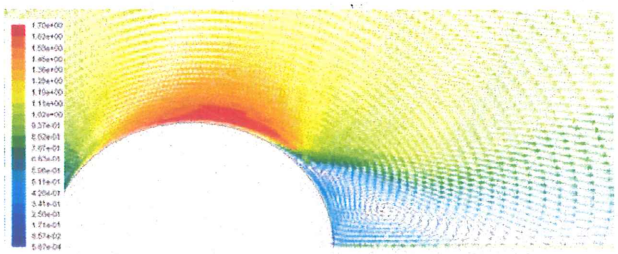
- Discretization problem

For the airfoil study, the aim will be to compare the numerical results provided by *Fluent* with the analytical calculation. These theoretical transformations do not take any viscous effects into account. Therefore, the simulation was run under inviscid conditions.
In this ideal frictionless case, the flow around the sphere remains smooth and undisturbed by the geometry. The pressure coefficient Cp is represented as a function of the sphere angle θ. Positive values of Cp indicate high pressures while negative values indicate low pressures. As we can see on the graph, the pressure is very high at the stagnation point and consequently initiating a drag force. However, it is cancelled by the identical negative drag occurring at the back of the sphere. To conclude, in these theoretical conditions, the sphere does not produce any drag at all.

Under these considerations, a major problem appeared in our first simulation. Despite the inviscid model, the results appeared to underline a major contradiction with our previous conclusions. The figure below represents the velocity vectors of the flow:

Discretization scheme

- Pressure : Standard
- Pressure & Velocity: SIMPLE
- Momentum : 1st Order Upwind



The resulting velocity distribution is very surprising since we were not expecting to have any disturbances in the flow field. Flow separation is due to friction, which causes the streamlines to reverse. The separation point is located on the second half of the sphere, resulting in the formation of a separation bubble at the back. Another consequence of this is that the flow field surrounding the sphere is no longer symmetrical. The region of separated flow is dominated by unsteady, recirculating vortices that create a wake and consequent drag. Our first assumption is to consider that the

discretization schema is not adapted to this case of study. Therefore, each of the inputs needs to be investigated independently. Consequently, we will try to select the best fitted schema for our problem.

a) Choice of the Solver

The discretization scheme's characteristics we need to enter in ***Fluent*** depend on the solver we apply to the Euler system. There exist two different types: **'Segregated'** or **'coupled'**. These two approaches differ in the way the continuity, momentum, energy and species equations are solved.

- Coupled solver

The **coupled solver** solves the equations of continuity, momentum, energy and transport simultaneously as a system of governing equations. The system can be written as a single vector as follow:

$$\frac{\partial}{\partial t}\int_V W\,dV + \oint [F-G]\cdot dA = \int_V H\,dV \qquad with:\ W=\begin{Bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho E\end{Bmatrix},\ F=\begin{Bmatrix}\rho v\\ \rho vu+p\hat{i}\\ \rho vv+p\hat{j}\\ \rho vw+p\hat{k}\\ \rho vE+pv\end{Bmatrix},\ G=\begin{Bmatrix}0\\ \tau_{xi}\\ \tau_{yi}\\ \tau_{zi}\\ \tau_{ij}v_j+q\end{Bmatrix},\ H \leftrightarrow source\ terms$$

Where: $\begin{cases}\rho : density & p : pressure\\ v : velocity & \tau : viscous\ stress\ tensor\\ E : energy & q : heat\ flux\end{cases}$

We notice that there are 6 unknowns for only 5 equations. The equation relating the energy E and the enthalpy H is necessary to close the system, so that we have: $E = H - {}^{p}/_{\rho}$ *where*: $H = h + |v|^2/2$

The coupled set of governing equations is then discritized in time by using an explicit or an implicit time marching solution, until a steady state solution is achieved.

- Segregated solver

According to the required order of accuracy, the *'order of the upwind scheme'* needs to be selected. This 1st parameter influences the approximation technique applied to calculate the quantities across the cells.
The segregated solver solves the equations sequentially. First of all the momentum equation is disretized. The aim is to solve the velocity field but the pressure field and mass fluxes at the face are unknown. The pressure values are needed for this first resolution step, which leads us to adopt a *'Pressure Interpolation Scheme'*.
Finally, since the segregated solver does not solve the equations simultaneously, the pressure and velocity fields have to be coupled so that their inter-dependency is taken into account during the resolution. Therefore, the *'pressure/velocity coupling'* is the last parameter to fix.
Both solvers are now applicable to a broad range of flows. Originally, the segregated solver was used for incompressible and average compressible flows, and the coupled solver strongly favored for high speed compressible flows. Now days, the coupled would be best suited if strong body forces intervene, causing the flow to be highly coupled. In our study it is not the case, and taking into consideration that a coupled implicit solver takes up more memory, we choose to use the **segregated method**.

- Order of Upwind scheme

When **1st order upwind** scheme is applied, the quantities at each cell face are held constant across the face and equal to the cell average value of the upstream cell. This approximation greatly simplifies the calculation.
However, if a higher order of accuracy is desired, a **2nd order upwind** scheme is needed. The cell quantities are determined by using a multidimensional linear reconstruction approach. The value ϕ_f is determined by using the cell centered value of the upstream cell and its gradient as follow:

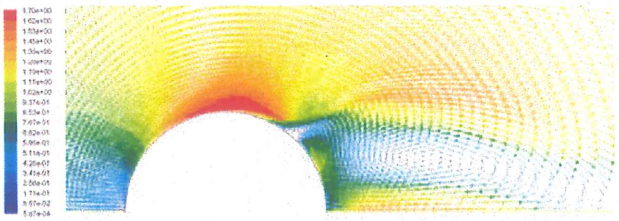
$$\phi_f = \phi + \nabla\phi \cdot \Delta\vec{s} \qquad \begin{cases} \phi : \textit{Quantity at the upstream cell center} \\ \nabla\phi : \textit{Gradient of this quantities upstream} \\ \Delta\vec{s} : \textit{displacement vector between both cells} \end{cases}$$

The gradients are computed by using the divergence theorem. Its expression in discrete form is the following: $\nabla\phi = \frac{1}{V}\sum_{f}^{N_{faces}} \tilde{\phi}_f \vec{A}$.

The computational effort required in this case is greater than for the 1st order scheme so the choice as for which to use has to be made by taking into account both the computational capacity and the desired level of accuracy.

A simulation was run by using a 2nd order upwind scheme; however this modification alone did not affect the presence of separated flow. The following velocity vector field was obtained, on which the separation bubble still appears:

- Pressure : Standard
- Pressure & Velocity: SIMPLE
- Momentum : 2nd Order Upwind



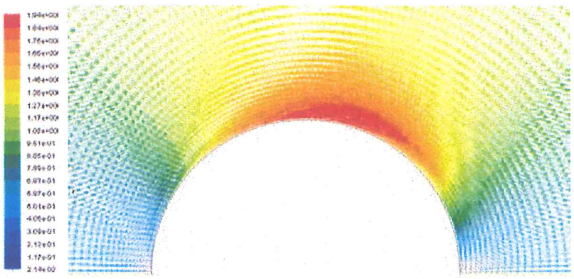
If we compare both results obtained up to now, the velocity field is nevertheless different with a higher pressure appearing at the separation point and a modified velocity field downstream of the this point. Considering that this parameter affects the result we decide to use a **2nd order upwind scheme** for all the following simulations.

- Pressure Interpolation schemes

The default scheme **'Standard'** in *Fluent* considers the momentum equation coefficients used to interpolate the pressure values. This procedure is valid as long as the pressure variation between cell centers is smooth. If the geometry produces jumps or large gradients between the control volumes it produces a high gradient of pressure profile at the cell face. Consequently, the discrepancy increases greatly and large overshoots/undershoots of cell velocity occur.
Another assumption made by the standard scheme could explain the incoherence of our result. Fluent assumes that the normal pressure gradient at the wall is zero which is valid for boundary layers. However it is not the case in the presence of curvature, characteristic of a spherical geometry. These false pressure gradients at the wall lead to inaccurate velocity distributions along the wall.

determine the staggered pressure across a staggered control volume. This interpolation technique is recommended for flows in highly curved domain. This scheme seems particularly adapted for a sphere and we run a new simulation using this pressure interpolation method.

- Pressure : PRESTO !
- Pressure & Velocity: SIMPLE
- Momentum : 1nd Order Upwind



The result is very satisfying. Indeed, the separation region has disappeared and the flow no longer separates on the second half of the sphere. The velocity field corresponds to what we expect of an inviscid model, which leads us to apply the **PRESTO!** interpolation method for the next simulations.

- Pressure / Velocity coupling

If we consider the Navier-Stockes set of equations, a linear dependence between pressure and velocity exists, and vice versa. A specific treatment is required to formulate this pressure velocity coupling. Several methods exist for a segregated solver.

The default method provided by Fluent is the **SIMPLE** algorithm. For relatively uncomplicated problems, a converged solution can be obtained faster by using the **SIMPLEC** method. Indeed, the pressure correction factor is usually set to 1, which speeds up the convergence. We tried the two pressure/velocity coupling possibilities:

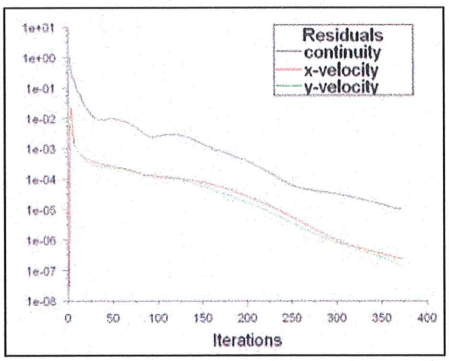


Figure 1: SIMPLE coupling

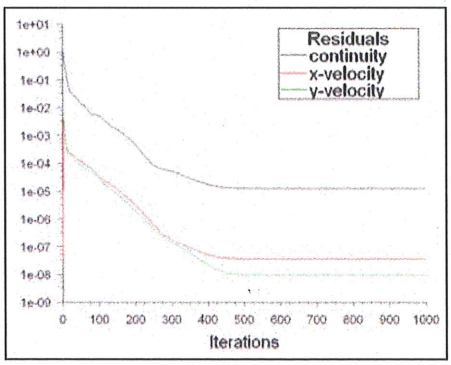


Figure 2: SIMPLEC coupling

After running several simulations for both algorithms, the residuals behaviour remained unchanged. By comparing these two figures we observe that the velocity residuals decrease at the same rate. For the SIMPLEC case, the continuity residual reaches a constant value larger than 10^{-5} whereas it continues to decrease for the SIMPLE coupling. The SIMPLEC coupling method does not accelerate the convergence in our case consequently we will continue to use the SIMPLE coupling.

To conclude, this is the final solver and its discretization we have chosen to adopt:

Segregated solver → Pressure → *PRESTO*; Pressure & velocity → *SIMPLE*; Momentum → 2^{nd} *order upwind*

1) Grid effect

In order to determine what the most suitable grid is, we undertake a grid effect study. The number of nodes is increased until the mesh has no more effect on the results. The mesh just at the limit is optimum for our study since it is both accurate and economic.
Down below are three different meshes we used for the calculations, where the number of nodes goes from weak to moderate and finally high.

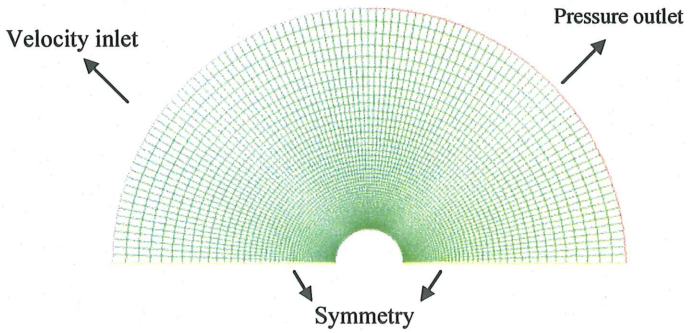
Number of nodes	Mesh density
4131	Very weak (1)
9301	Weak (2)
16281	Moderate (3)
32481	High (4)

The parameter we choose for comparing these grids is the drag coefficient: $C_D = \dfrac{D}{1/2\,\rho V^2 S}$

In this 2D case, the sphere's frontal surface is equal to unity. Air has been used as the working fluid and therefore the density is known. The drag force D has been monitored by FLUENT and we easily access its value corresponding to the integral of the forces in the X direction. Considering that the viscous model is inviscid, the value of this coefficient is close to 0.

$$\left.\begin{array}{l}\rho_{water} = 1.225\ kg/m^3 \\ V = 1\ m/s \\ S = 1\ m^2\end{array}\right\} \begin{array}{lll} D_{\_1} = 0.001008 & \Rightarrow & 1.645\times10^{-3} \\ D_{\_2} = 0.000645 & \Rightarrow & 1.053\times10^{-3} \\ D_{\_3} = 0.000597 & \Rightarrow & 9.747\times10^{-4} \\ D_{\_4} = 0.000589 & \Rightarrow & 9.616\times10^{-4} \end{array}$$

We can conclude that the mesh (1) is not at all sufficient enough since the result is very far from the correct value. Then it is a matter of how precise the results need to be according to the computational effort available. We will from now on continue with mesh (3) which provides us with an accurate result. Indeed, any further increase in mesh density hardly affects the result and can be considered as useless for our study. We have reached a state where the calculation is **grid independent**. Therefore the chosen mesh is composed of **16281 nodes.**



1) Numerical results

a) Velocity field

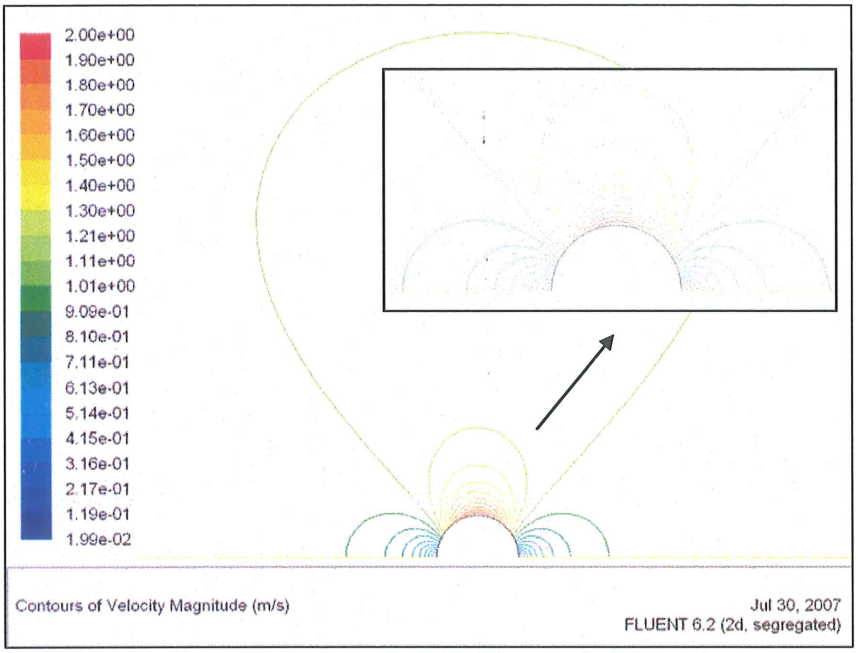
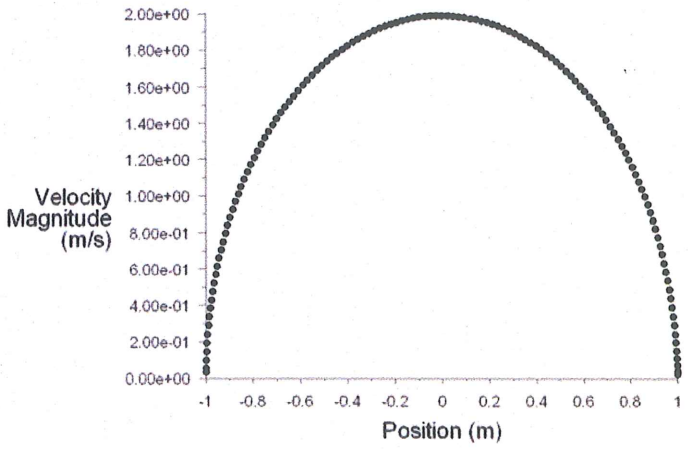


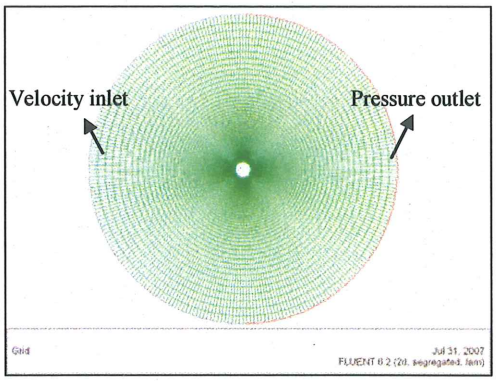
Figure 1: Contours of velocity magnitude

As predicted previously, very high pressures are located at $\theta = 0°$ and 180° corresponding to stagnation points. The velocity contours around those two regions are identical which confirms that hardly any drag is generated by the body due to this cancellation.
As for the velocity behaviour across the chord, the sphere's curvature causes the flow stream to accelerate on the first half, up to 90°, and naturally decelerate downstream. The following plot illustrates the velocity's evolution along the chord.

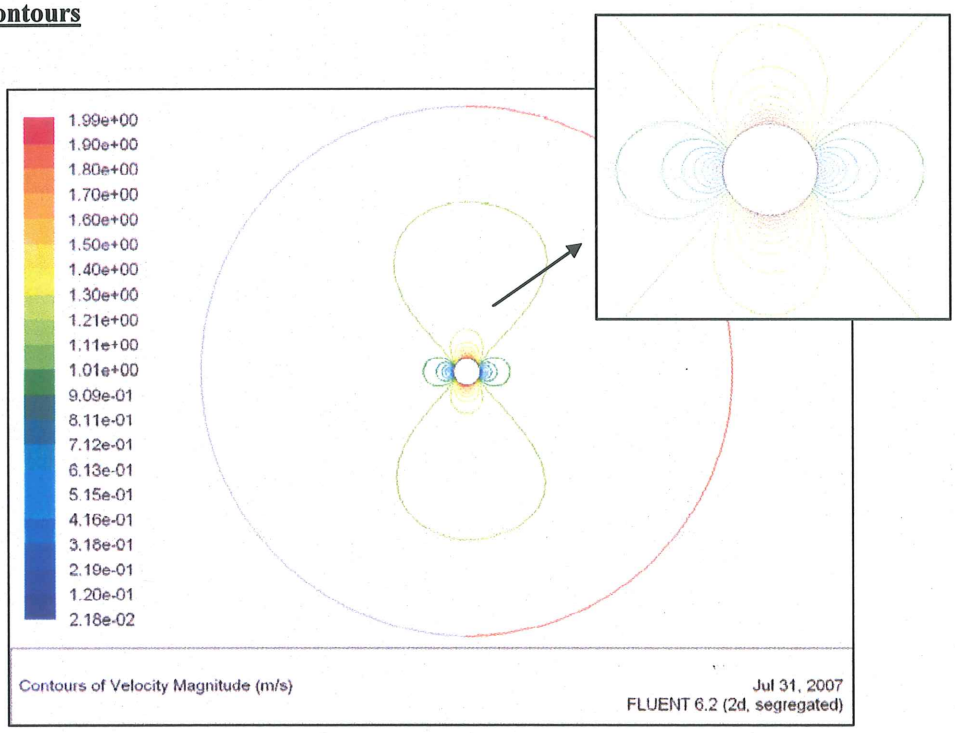


1) Symmetrical study

Previously we used the geometry's symmetrical property to naturally assume that its impact on the surrounding flow would be symmetrical as well. We will now illustrate this by applying an identical simulation to the whole circle. We keep the same mesh, repeated on the other half, as well as the previous parameters. Here is the grid we generated for this study:



Velocity contours



The velocity distribution appears to be identical to the half sphere. In addition, the symmetrical properties of the inviscid calculation applied to a sphere are confirmed, since there is no difference in velocity contours between the top and bottom half of the geometry.

APPENDIX D

Coord.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%% The objective of this program is to generate as an %%%%%%%%%%%%
%%%%%%%%% output the coordinates of the 4-digit airfoil NACA 4412 %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Calculation of the coordinates

c = 1;                          % Unit chord
N = 500;                        % Number of points calculated on each surface

f = 0.039995;                   % maximum camber
xf = 0.395;                     % position of maximum camber
t = 0.120035 / c ;              % thickness/chord

x1 = (xf/c);

% Equations for the upper and lower coordinates
for i = 1:N

    % Horizontal increment
    dx = c / (N-1);
    x(i) = dx * (i-1);

    % Camber line
    if ((x(i)/c) <= x1)
        yc(i) = ((f/c)*(1/(x1^2))*(2*x1*(x(i)/c) - (x(i)/c)^2))*c;
        dyc(i) = ((f/c)*(1/(x1^2))*(2*x1*(1/c) - 2*(x(i)/c)))*c;
    else
        yc(i) = ((f/c)*(1/(1-x1)^2)*((1-2*x1)+2*x1*(x(i)/c)-(x(i)/c)^2))*c;
        dyc(i) = ((f/c)*(1/(1-x1)^2)*(2*x1*(1/c)-2*(x(i)/c)))*c;
    end

    % Thickness distribution
    yt(i) = (5*t*(0.29690*x(i)^0.5 - 0.12600*x(i) - 0.35160*x(i)^2 + 0.28430*x(i)^3 -
0.10150*x(i)^4))*c;

    % Angular distribution
    teta(i) = atan(dyc(i));

    % Upper and lower coordinates
    y_upper(i) = yc(i) + yt(i)*cos(teta(i));
    y_lower(i) = yc(i) - yt(i)*cos(teta(i));
    x_upper(i) = x(i) - yt(i)*sin(teta(i));
    x_lower(i) = x(i) + yt(i)*sin(teta(i));
end

N_LE = 8;
for i = 1:N_LE
    dx_LE = 0.002 / (N_LE-1);
    x_LE(i) = dx_LE * (i-1);
    yc_LE(i) = ((f/c)*(1/(x1^2))*(2*x1*(x_LE(i)/c) - (x_LE(i)/c)^2))*c;
    dyc_LE(i) = ((f/c)*(1/(x1^2))*(2*x1*(1/c) - 2*(x_LE(i)/c)))*c;
    yt_LE(i) = (5*t*(0.29690*x_LE(i)^0.5 - 0.12600*x_LE(i) - 0.35160*x_LE(i)^2 +
0.28430*x_LE(i)^3 - 0.10150*x_LE(i)^4))*c;
    teta_LE(i) = atan(dyc_LE(i));
    y_upper_LE(i) = yc_LE(i) + yt_LE(i)*cos(teta_LE(i));
    y_lower_LE(i) = yc_LE(i) - yt_LE(i)*cos(teta_LE(i));
    x_upper_LE(i) = x_LE(i) - yt_LE(i)*sin(teta_LE(i));
    x_lower_LE(i) = x_LE(i) + yt_LE(i)*sin(teta_LE(i));
end

fid = fopen('E:\sophia\ITU\Viscous NACA 4412\FINAL\journals\NACA4412_LE_upper.txt','w')

fprintf(fid, '/ Journal file for GAMBIT\n')
fprintf(fid, '/ Identifier NACA 4412, leading edge upper\n\n\n')

zcoord_LE=0*ones(1,N_LE);
VECT = [x_upper_LE;y_upper_LE;zcoord_LE];
fprintf(fid,'vertex create coordinates      %6.6f      %6.6f      %6.6f\n',VECT);
fclose(fid)
```

```
fid = fopen('E:\sophia\ITU\Viscous NACA 4412\FINAL\journals\NACA4412_LE_lower.txt','w')

fprintf(fid, '/ Journal file for GAMBIT\n')
fprintf(fid, '/ Identifier NACA 4412, leading edge lower\n\n\n')

zcoord_LE=0*ones(1,N_LE);
VECT = [x_lower_LE;y_lower_LE;zcoord_LE];
fprintf(fid,'vertex create coordinates      %6.6f      %6.6f        %6.6f\n',VECT);
fclose(fid)

%% GAMBIT journal file containing the airfoil coordinates (500 points)used

fid = fopen('E:\sophia\ITU\Viscous NACA 4412\FINAL\journals\NACA4412_upper.txt','w')

fprintf(fid, '/ Journal file for GAMBIT\n')
fprintf(fid, '/ Identifier NACA 4412, upper surface\n\n\n')

zcoord=0*ones(1,N);
VECT = [x_upper;y_upper;zcoord];
fprintf(fid,'vertex create coordinates      %6.6f      %6.6f        %6.6f\n',VECT);
fclose(fid)

fid = fopen('E:\sophia\ITU\Viscous NACA 4412\FINAL\journals\NACA4412_lower.txt','w')

fprintf(fid, '/ Journal file for GAMBIT\n')
fprintf(fid, '/ Identifier NACA 4412, lower surface\n\n\n')

VECT = [x_lower;y_lower;zcoord];
fprintf(fid,'vertex create coordinates      %6.6f      %6.6f        %6.6f\n',VECT);
fclose(fid)
```