POLITECNICO DI MILANO

SCUOLA DI INGEGNERIA INDUSTRIALE E
DELL'INFORMAZIONE

# Aerodynamics of a glider: development of computational tools and application studies

*Autore:*
Tommaso Guggiari
Matricola 817320

*Relatore:*
Prof. Lorenzo Trainelli
*Correlatore:*
Ing. Giovanni Droandi

*Tesi di Laurea Magistrale in Ingegneria
Aeronautica*

*Anno Accademico 2015-2016*

# Ringraziamenti

# Abstract

Questa tesi presenta la creazione di un modello aerodinamico per l'aliante V5 "Rondone", un progetto dell'Ing. Vittorio Pajno. Lo strumento principale utilizzato in questa analisi è il software COMPA, sviluppato presso il Dipartimento di Scienze e Tecnologie Aerospaziali del Politecnico di Milano. Il software open source Xflr5 è stato utilizzato per completare questa analisi con i contributi viscosi e instazionari.

Come scopo secondario di questo progetto di tesi, i codici per l'analisi del V5 sono stati generalizzati per essere applicabili a un'ampia gamma di velivoli. Gli strumenti sviluppati consentono la generazione automatica di una robusta griglia di calcolo per COMPA e la sua validazione a partire da un elastico set di parametri descrittivi della geometria di un velivolo convenzionale.

# Abstract

This thesis presents the creation of an aerodynamic model for the sailplane "V5 Rondone", a design by Mr. Vittorio Pajno. COMPA, a 3D panel method software developed at the Department of Aerospace Science and Technology of Politecnico di Milano, was the main tool for this analysis. The open source software Xflr5 was employed to refine the results with the missing viscous and unsteady terms.

As a secondary result of this project the computational set-up for the the V5 analysis has been generalised to allow the analysis of a wide range of aircrafts. A framework was created that allows the automatic generation and validation of a robust mesh for COMPA from an elastic set of parameters describing a monoplane geometry.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **AoA** | **A**ngle **o**f **A**ttack |
| **AoS** | **A**ngle **o**f **S**ideslip |
| **BC** | **B**oundary **C**ondition |
| **CS** | **C**ertification **S**tandard |
| **EASA** | **E**uropean **A**viation **S**adety **A**organisation |
| **EAS** | **E**quivalent **A**ir **S**speef |
| **ICB** | **I**ndividual **C**omponents **B**uild-up |
| **LLT** | **L**ifting **L**ine **T**heory |
| **VLM** | **V**ortex **L**attice **M**ethod |

# List of Symbols

| | | |
|---|---|---|
| $a$ | Parameter of $g_0$ distribution function | [-] |
| $a_c$ | Weight of the cosine function | [-] |
| $b$ | Parameter of $g_0$ distribution function | [-] |
| $c$ | Aerodynamic chord | [m] |
| $C_D$ | Drag coefficient | [-] |
| $C_{Di}$ | Induced drag coefficient | [-] |
| $C_{Dv}$ | Viscous drag coefficient | [-] |
| $C_{D0}$ | 0-Lift drag coefficient | [-] |
| $C_f$ | Skin friction coefficient | [-] |
| $C_{fe}$ | Equivalent skin friction coefficient | [-] |
| $C_L$ | Lift coefficient | [-] |
| $C_{Lmax}$ | Maximum lift coefficient | [-] |
| $C_{Mx}$ | Moment coefficient around the $x$ axis | [-] |
| $C_{My}$ | Moment coefficient around the $y$ axis | [-] |
| $C_{Mz}$ | Moment coefficient around the $z$ axis | [-] |
| $C_{X,Y}$ | Derivative of generic x coefficient on generic Y angle | [-] |
| $C_X\|_{trim}$ | Trim value of generic $C_X$ | [-] |
| $C_{x_i}$ | Generic aerodynamic coefficient | [-] |
| $C_{x_{MaxPan}}$ | Aerodynamic coefficient with the highest grid refinement | [-] |
| $C_{x_{MaxPan}}$ | Aerodynamic coefficient with the highest grid refinement | [-] |
| $C_Y$ | Sideslip force coefficient | [-] |
| $d$ | Lever-arm of the Lift on the centre of gravity | [m] |
| $D_H$ | Horizontal fuselage diameter | [m] |
| $D_H$ | Vertical fuselage diameter | [m] |
| $f$ | Lifting body span-wise distribution function | [-] |
| $\overline{f}$ | Lifting body span-wise distribution function | [-] |
| $f_0$ | Lifting body span-wise distribution function c | [-] |
| $FF$ | Form function | [-] |
| $g_0$ | Lifting body span-wise distribution function | [-] |
| $H_0$ | Fuselage geometry centroid $z$ offset | [m] |
| $k_e$ | Thickening parameter | [-] |
| $m$ | Number of panels/grid points on the airfoil surface | [-] |
| $M$ | Number of control surfaces in a lifting body | [-] |
| $n$ | Span-wise number of panels/grid points | [-] |
| $N$ | Number of sections in a body | [-] |
| $\{p\}$ | Vector of the weights of the thickening centres | [-] |
| $p_l$ | Left thickening weight of $f_0$ | [-] |
| $p_r$ | Right thickening weight of $f_0$ | [-] |
| $Q$ | Interference factor | [-] |

| | | |
|---|---|---|
| $s$ | Normalised arc-length coordinate | [m] |
| $S$ | Reference surface | [m$^2$] |
| $S_{wet}$ | Wet surface | [m$^2$] |
| $\{s_u\}$ | Uniform distribution between $0$ and $1$ | [-] |
| $t$ | Thickness | [m] |
| $x_A$ | Aerodynamic centre $x$ coordinate | [m] |
| $x_G$ | Centre of gravity $x$ coordinate | [m] |
| $\{x\}$ | Vector of $x$ coordinate points (2D geometry) | [m] |
| $v_e$ | Equivalent air speed | [m/s] |
| $[x]$ | Matrix of $x$ coordinate points (3D geometry) | [m] |
| $\{y\}$ | Vector of $y$ coordinate points (2D geometry) | [m] |
| $[y]$ | Matrix of $y$ coordinates points (3D geometry) | [m] |
| $W$ | Weght | [N] |
| $\{z\}$ | Vector of $z$ coordinate points (2D geometry) | [m] |
| $[z]$ | Matrix of $z$ coordinates points (3D geometry) | [m] |
| $z_A$ | Aerodynamic centre $z$ coordinate | [m] |
| $z_G$ | Centre of gravity $z$ coordinate | [m] |
| $\alpha$ | Angle of attack | [rad or °] |
| $\beta$ | Angle of sideslip | [rad or °] |
| $\Delta_A$ | Aileron deflection | [rad or °] |
| $\Delta_E$ | Elevator deflection | [rad or °] |
| $\Delta_R$ | Rudder deflection | [rad or °] |
| $\varepsilon$ | Error | [-] |
| $\varepsilon_{Cx_i}$ | Relative error for the grid validation | [-] |
| $\{\xi\}$ | Vector of the coordinates of the thickening centres | [-] |

# Chapter 1

# Introduction

## 1.1 Motivation

The inspiration behind this project comes from Vittorio Pajno, an aeronautical engineer currently working on the design of the sailplane V5 "Rondone".

Mr. Pajno is an old school engineer and based his hand-made drawings on his life-long experience and his vast theoretical and practical knowledge of the field. In the project of the V5 he followed Roskam-type methods [1] and existing regulation requirements [2] for sizing the different parts of the aircraft and estimating the performance indicators.

EASA regulations requires the sailplane to comply with the CS-22 airworthiness code. The code includes the formulae for computing the relevant parameters during the design phase and their compliance range. After a prototype is built, test flights have to be carried out to demonstrate the compliance and to verify the relevant parameters of the Pilot Operating Handbook [3].

The design of sailplanes is an established science and actual flight qualities and performance are not expected to fall far from predictions. Nevertheless, given the small scale of this project, it is important that minimal alterations to the project may be needed after the flight tests. For this reason Mr. Pajno proposed a collaboration with Politecnico di Milano aimed at verifying the flight qualities of the glider with adequate accuracy.

Following the input from Mr. Pajno it was decided to start this thesis project, the primary aim being the numerical estimation of the

V5 flight characteristics using the internal resources of Politecnico di Milano.

The software COMPA was chosen as the main tool for this work. COMPA is a state-of-the-art 3D panel method software for computing the aerodynamic coefficients on an arbitrary body at given aerodynamic angles. Its previous usage has been limited although it has been already tested and validated with experimental results [4]. One of the reasons behind the choice of COMPA was the opportunity to test the consistency of its results when applied to a conventional full-scale glider configuration with deflectable control surfaces.

For its application, COMPA needs a pre-processor that generates the mesh and a post-processor to elaborate the coefficients in different cases in order to compute the aerodynamic model.

During the course of the project, its scope was widened to include the writing of a set of codes for the following tasks:

- Generation of a robust COMPA mesh for any given motor-less aircraft geometry with control surfaces, described by a set of parameters and airfoil coordinates.

- Convergence analysis and validation of each part of the mesh through multiple iterative simulations with COMPA in different conditions.

- Trim of the aircraft for the conditions to be tested through a preliminary COMPA analysis.

- Post-processing of the coefficients for the computation of the aerodynamic database.

After this part was completed, it was chosen to complement the work on the V5 with another set of simple analyses using the open source software Xflr5 to overcome the steady and inviscid nature of COMPA results and achieve the following:

- A viscous drag contribution estimation to complement the inviscid results from the COMPA analysis

- An analysis with unsteady boundary conditions, for the evaluation of the aircraft derivatives with respect to the aerodynamic angular rates.

- A comparison with the COMPA results in steady inviscid analyses for validation purposes.

## 1.2   State of the art and related works

There are many different methods that can be used to predict the aerodynamics of an aircraft. The simplest one being manual calculations of aerodynamic theories and empirical formulae, such as those contained in Datcom [5] and in Airplane Design [1] by Roskam.

Semi-empirical methods are very popular in the aircraft industry for performance predictions in the preliminary design phase. These codes are mostly based on simple potential-flow methods and empirical formulae mainly coming from Datcom. Aeronautical companies often develop their own codes based on their experimental results, but they do not release them to the public. An example of the implementation of a semi-empirical method based on this approach can be found in Ref. 6.
Some of the modern available software based on semi-empirical methods includes DATCOM+ Pro (based on DATCOM+ which was in turn based on Digital Datcom) and Advanced Aircraft Analysis. Ref. 7 features DATCOM+ Pro, used to model a UAV aircraft

Aerodynamic potential-flow codes currently used for aerodynamics prediction of full-size aircraft feature mainly vortex-lattice (VLM) and 3D panels methods. Strip theory and lifting line theory (LLT) are simple methods still used in some instances, but are now becoming outdated due to the increase of the processing power in modern computers.

Due to its simplicity and short computation times, VLM is a very popular tool for the preliminary design phase, along with semi-empirical methods. It is implemented in some of the most widespread aerodynamic software available, such as Athena Vortex

Lattice(AVL), Tornado [8] and Xflr5 [9]. An approach featuring AVL together with Datcom+ is used in Ref. 10 to find the stability and control derivatives of a sailplane, while in Ref. 11 Tornado is used as the aerodynamic module for the optimisation of a glider.

3D panel methods offers a higher fidelity thanks to the accurate representation of the thickness of three-dimensional wings. If implemented correctly they can compute very accurate solutions for subsonic attached flows where friction drag can be neglected. In Ref. 12 the author uses FastAero, a 3D panel code developed at MIT, to simulate the aerodynamics of a sailplane, and compares the results with VLM and strip theory.

The main issue with potential-flow methods is their neglecting of the viscosity. This leads to substantial errors in the calculation of the total drag and the prediction of stall behaviour. Some authors have proposed a way to develop a viscous quasi three-dimensional aerodynamic solver for lifting surfaces that couples a 2D viscous airfoil code or data with a simple 3D potential-flow method [13]. Ref. 14 presents an iterative method for computing the total drag and capturing the non-linear lift behaviour on a finite length wing by coupling an airfoil wind tunnel data with a LLT implementation.

Ref. 15 uses the following codes developed internally at the *University Federico II of Naples* to compute the aerodynamic database of a light aircraft: a custom 3D panel method code, the semi-empirical code AEREO and the code NLWING featuring a viscous correction of the LLT with a set of polars from a 2D viscous code.

Ref. 16 features the 3D panel software COMPA for the aerodynamic analysis and optimization of a tailless sailplane.

The work performed in this thesis uses state-of-the art 3D panel method software with a great focus on the refinement of the mesh and the achievement of an optimal panel distribution. For this reason it aims at achieving a better accuracy in the computation of the aerodynamic database than the works cited in this section. The aerodynamic analisys is similar to that presented in Ref. 16, which features a simpler body geometry.

The unsteady part of the analysis was performed using VLM and is expected to achieve a degree of accuracy similar to that of the other VLM based approaches presented in this section. The viscous correction using LLT is similar to the one presented in Ref. 15.

## 1.3 Software and Methods

### 1.3.1 COMPA

COMPA (COMPlete Aircraft aerodynamic computation) is an aerodynamic software written and developed at the Department of Aerospace Technology and Science and technology of Politecnico di Milano by Giovanni Droandi and prof. Giuseppe Gibertini.

COMPA features is a three-dimensional low-order velocity based panel method for steady, inviscid, incompressible flows. It is based on the classical Hess and Smith method modified with the idea proposed by Maskew and Woodward [4, 17]. The code is capable of analysing both lifting and non-lifting body parts. It represents the outer surface of each part as a distribution of constant source panels and the mean line of the lifting parts as a distribution of vortex rings with horse-shoe vortices spawning from the trailing edge to represent the wake (aligned with the asymptotic velocity). The boundary conditions are used to enforce impermeability, with a no penetration Dirichlet condition on each panel centroid. The lift generation on lifting bodies is achieved with the enforcement of the Kutta condition as BC on the wake. For the lifting surfaces the source distribution is enforced to be symmetric with respect to the mean surface in order for the number of unknown sources and vortices to match that of the boundary conditions.

### 1.3.2 Xflr5

Xflr5 is an open source software suite that puts together different potential-flow methods for the analysis of an airplane or a finite-length wing. Its peculiarity is the integration with Xfoil,

which is used as a tool for a quasi-3D viscous analysis with the methods described in Ref. [13].

Xflr5 is used in this work with the purpose of completing the COMPA analises of the V5 with the computation of the unsteady terms of the aerodynamic model and the estimation of the lifting surfaces viscous drag. For the first purpose an unsteady VLM analysis of the full plane is performed and its steady results compared to those of COMPA for validation. For the second purpose an iterative viscous LLT analysis analogous to that described in Ref. 14 is implemented.

### 1.3.3 Semi-empirical drag relations

Semi-empirical relations from various authors [1, 18, 19, 20] were used to estimate the viscous drag contribution to compare with Xflr5 results and to provide the fuselage viscous drag term, which Xflr5 does not compute.

## 1.4 Outline of the Thesis

**Chapter Two: COMPA pre-processor**  Describes the development of a COMPA pre-processor to generate a generic sailplane mesh in its various parts. A particular emphasis is put in the creation of suitable panel distribution functions for each part, the inclusion of deflectable control surfaces and the creation of smooth, point-by-point junctions between the different blocks.

**Chapter Three: COMPA analysis**  Describes the mesh convergence analysis and validation. Then it reports the first preliminary analyses on the full aircraft geometry, used to compute the aerodynamic curves, the aerodynamic center and verify the linearity of the pitching moment with the elevator deflection. The chapter then describes the trim of the aircraft for specific flight conditions and the aerodynamic derivatives resulting from the main analysis on the trimmed sailplane are reported.

**Chapter Four: Viscous drag estimation**    Presents three alternative ways of estimating the viscous drag contributions: the preliminary design drag estimation, the zero-lift drag build-up from individual components semi-empirical approach and the computation of the drag of individual lifting surfaces with Xflr5 LLT method. The three methods are then compared and assessed and the final viscous drag polar and trimmed drag polar curves are presented.

**Chapter Five: Xflr5 unsteady analysis**    Describes the creation of an Xflr5 analysis using the VLM method to compute the unsteady part of the V5 aerodynamic model. The steady part of the analysis is used for validation by comparing it to COMPA results.

**Chapter Six: Conclusions and future developments**    Discusses the results and limitations of this thesis work and proposes future developments

## 1.5    The sailplane V5 "Rondone"

The V5 "Rondone" is a two seater motor-less sailplane designed by Vittorio Pajno and classified as a light-sport aircraft. The configuration of the glider is that of a high-wing, T-tail monoplane. It features a retractable single-wheel landing landing gear, aero brakes and optional ballast. The wing and horizontal empennage airfoils are custom designed by prof. Loek Boermans from Delft technical University.

Designed for training and sport flights, the peculiarity of the V5 is the seats configuration, which is side-by-side, whereas the traditional two seater motor-less glider design features the crew members seating one in front of the other. The advantage of this configuration is a better ease of communication between the crew members, which makes training easier and the sport flights more enjoyable. The downside is an increase of pressure drag, partly balanced by the reduction of wet surface compared with traditional configurations.

TABLE 1.1: V5 - Fuselage data.

| | |
|---|---|
| Length: | 1.50 m |
| maximum width | 1.05 m |
| maximum cross-surface | 1.35 m$^2$ |

TABLE 1.2: V5 - Wing data.

| | |
|---|---|
| Airfoil | DU04-140 |
| Span | 20 m |
| Surface | 11.973 m$^2$ |
| Aspect ratio | 33.41 |
| Taper ratio | 3.86 |
| Sweep angle (root to tip) | 1.45° |
| **Chords** Root Tip Mean geometric Mean aerodynamic chord | 810 mm 210 mm 597 mm 644 mm |
| **Dihedral angles** (root to tip) | 0° 3° 3.6° 4.5° 6.6° 13.2° |
| **Ailerons** Span Mean chord Surface | 4.00 m 78 mm 0.3119 m$^2$ |

TABLE 1.3: V5 - Horizontal Empennage data.

| | |
|---|---|
| Airfoil | DU86-137-25 |
| Span: | 3.33 m |
| Root chord | 500 mm |
| Tip chord | 270 mm |
| Surface | 1.335 m$^2$ |
| Volume ratio | 0.854 |
| Rudder surface | 0.333 m$^2$ |

TABLE 1.4: V5 - Vertical Empennage data.

| | |
|---|---|
| Airfoil | FX-L-150 |
| Span: | 1.50 m |
| Root chord | 1000 mm |
| Tip chord | 800 mm |
| Surface | 1.35 m$^2$ |
| Volume ratio | 0.887 |
| Rudder surface | 0.472 m$^2$ |
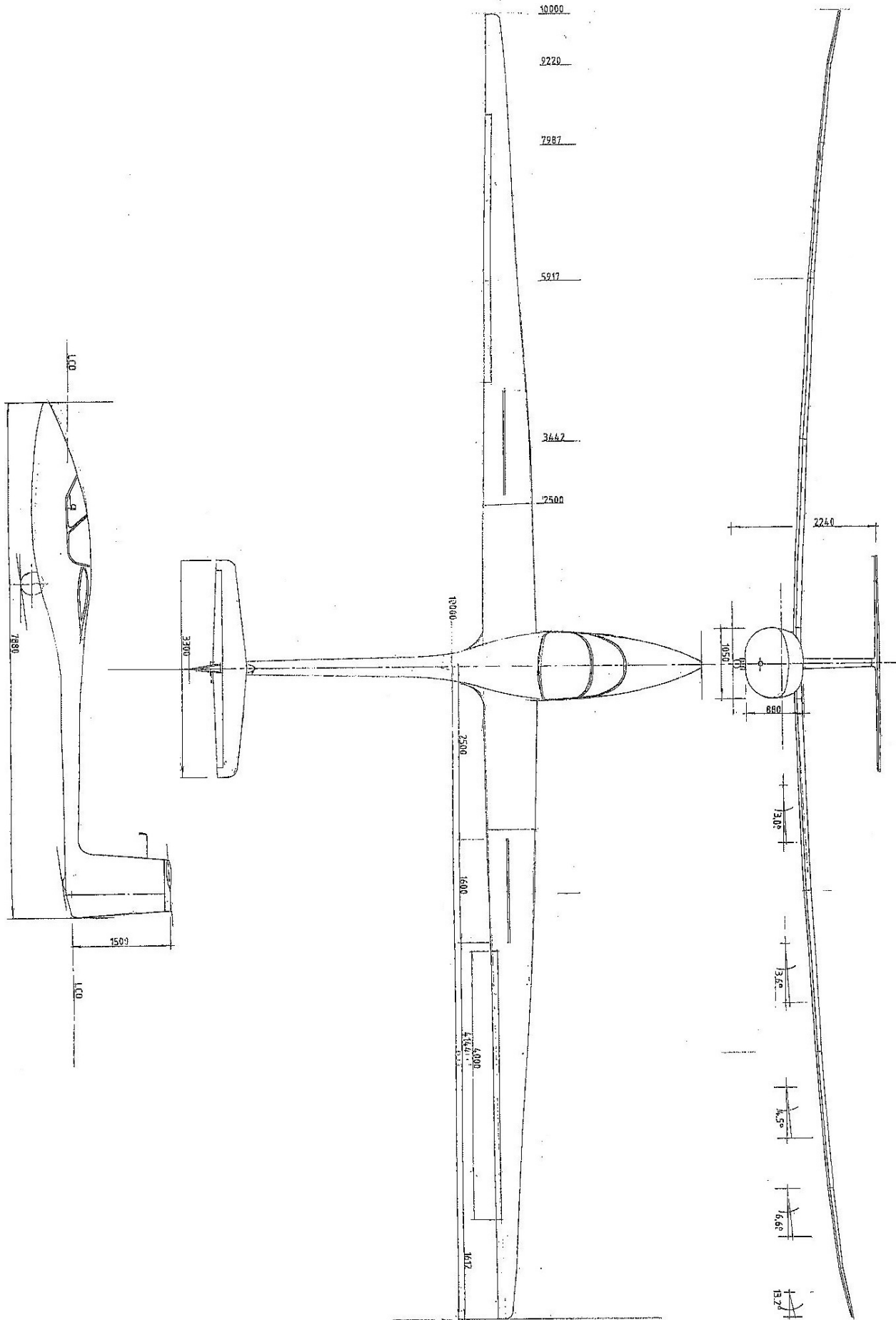
FIGURE 1.1: Three views of the V5 "Rondone".

# Chapter 2

# COMPA pre-processor

## 2.1   introduction

The COMPA pre-processor was implemented in Matlab. Its purpose is to generate the files COMPA needs as inputs. Those include the test case name, the aerodynamic angles for which to compute the coefficients and, most importantly, the mesh. All this data needs to be provided in the format accepted by the software and with all the necessary precautions to ensure numerical stability and robustness.

The mesh needs to be organized in blocks, each of which is a grid of $n \times m$ points in the 3D space, where $n$ is the number of span-wise grid lines (placed along the $y$ axis, longitudinal axis of the body in Figure 2.1) and $m$ is the number of points along the airfoil profile (placed on the $x - z$ plane in fig 2.1). The grid points are the edges of the $(n-1) \times (m-1)$ panels representing the outer surface of an aerodynamic body.
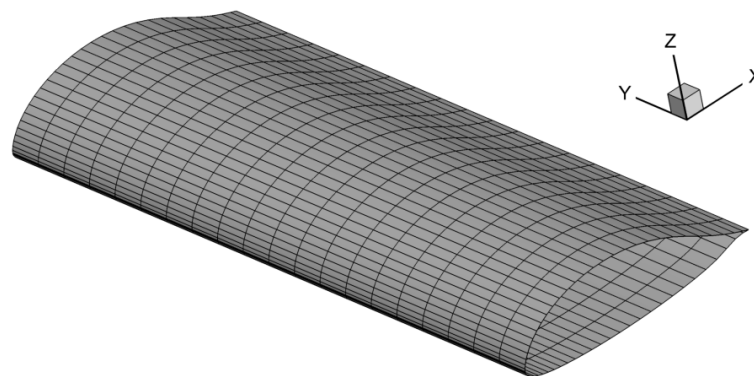
FIGURE 2.1: COMPA mesh example: a body made of a single block with 60x20 panels.

The mesh code was written in Matlab and is made by the following routines and subroutines:

- **Aircraft**: puts together the meshes of the single parts with coordinate shifts and rotations.

  - **Lifting surface**: Used to generate the wing and the empennages

    - **Airfoil**: Used by the lifting body subroutine to generate the airfoil grid in the transversal section of the lifting body.

  - **Fuselage**: Generates the fuselage.

    - **2D Fuselage geometry**: Used by the Fuselage subroutine to generate the 2D transversal section geometry of the fuselage.

  - **Junctions**: Various scripts that take care of linking the different parts together in a smooth way.

## 2.2   Airfoil

The airfoil subroutine creates the mesh for the 2D sections along the longitudinal axis of the wing, normalised on the chord length. It generates a single curved row of grid points in 2D space, starting and ending at the trailing edge.

The code re-interpolates the coordinate points of the airfoil received as input in order to make a mesh suitable for COMPA and adds a control surface if requested. Since the airfoil geometry is a closed curve on the $x - z$ plane, there is no one-to-one correspondence between its points $x$ and $z$ coordinates. For this reason the airfoil is treated as a parametric curve, function of its arc-length. By using this approach it is possible to generate the desired arc-length grid distribution and then compute the $x$ and $z$ coordinates of the mesh points via interpolation.

## 2.2.1 Panels distribution

These are the factors that needs to be taken into account for the subroutine to work well with any airfoil:

- COMPA uses the projection of the panels on the body mean surface to distribute the vortices and enforces a symmetric source distribution on the upper and lower airfoil sides. For this reason non-symmetric upper and lower panel distributions may cause problems.

- For numerical stability, the mesh must be smooth: changes in panels size must be gradual.

- In order to reduce the discretization error on the pressure coefficient, the mesh needs to be finer where its gradient is expected to be larger, such as on the leading edge.

- An incorrect representation of the airfoil geometry due to discretization errors leads to an error in the direction of the flow resulting from the application of the boundary. conditions. This is especially a problem on the trailing edge where the Kutta condition needs to be enforced. If there is a significant curvature on the trailing edge the mesh needs to be refined there.

An approach to a code that would automatically size the panels based on the local curvature was tried. The panels size was set be inversely proportional to the local curvature, so that the mesh would automatically be thicker on the leading edge and in any other part where the curvature is relevant. This approach had a very accurate representation of the geometry, but was discarded because the resulting panel distribution would often be irregular. Also, the importance of a "proper" geometry representation is only strict on the trailing edge because of the Kutta condition, so the resulting distribution was not optimal for COMPA.

It was then decided to use two custom distribution functions for the grid points placement, one for the forward part of the airfoil and one for the aft part.
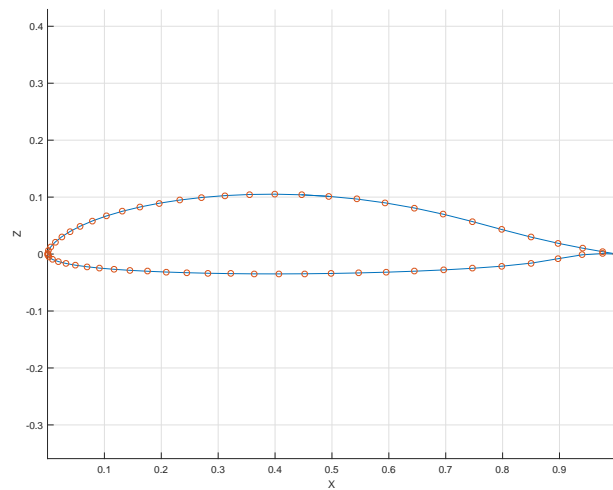
(a) DU04-140 (wing airfoil)
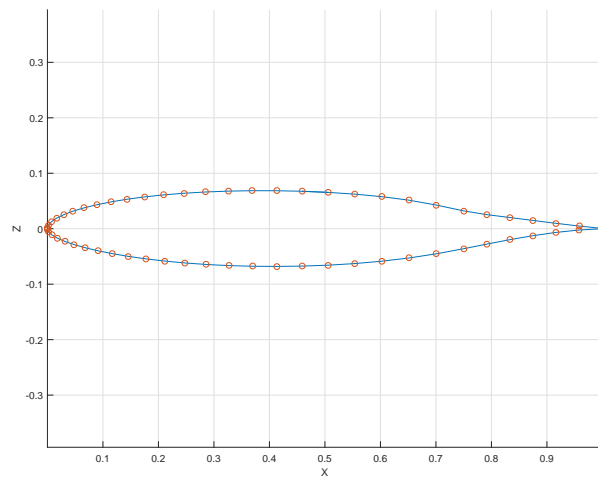


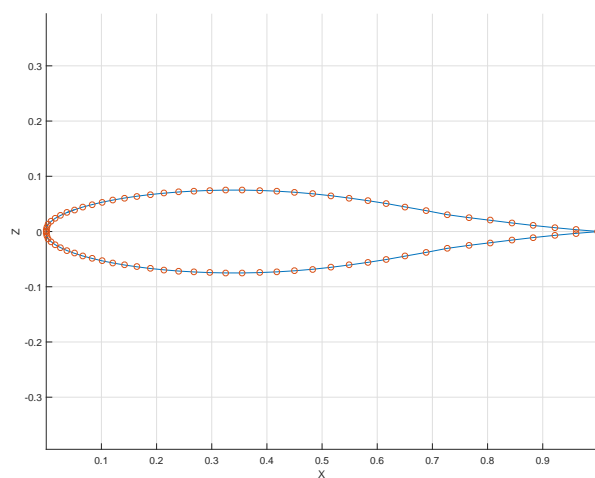(b) DU86-137-25 (horizontal empennage airfoil)



(c) FX-L-150 (vertical empennage airfoil)

FIGURE 2.2: V5 "Rondone" airfoil meshes.

The forward distribution function needs to size the top and bottom panels symmetrically, with a thickening of the grid on the leading edge:

$$\{s\} = (1 - \cos(\pi/2\{s_u\}))^{k_e} , \tag{2.1}$$

where $\{s\}$ is the vector of the normalized arc-length coordinates from the leading edge for both of the upper and lower side of the airfoil, $\{s_u\}$ is a uniformly spaced distribution between 0 and 1 and $k$ is a custom parameter for the user to increase or decrease the thickening on the leading edge.

It has been tested that a value of $k_e = 1$ works fine for common airfoils between 10% and 20% thickness and should be increased for a lower thickness and decreased for a higher thickness.

The aft mesh needs to be thickened near the trailing edge if there is a significant local curvature (Figure 2.2(a)), otherwise a uniform distribution works fine (Figures 2.2(b) and 2.2(c)), for this purpose the following distribution was implemented:

$$\{s\} = a_c[1 - \cos(\pi/2\{s_u\})] + (1 - a_c)\{s_u\}, \tag{2.2}$$

where $\{s\}$ is the vector of the normalized arc length coordinates from the trailing edge, used for both of the upper and lower side of the airfoil. The parameter $a_c$ is the weight of the cosine distribution. $a_c$ may be set higher than 0 to thicken the mesh on the trailing edge there

After tuning the parameters for the V5 airfoils, the resulting distribution from the 2.1 and 2.2 is shown in Figures 2.2(a), 2.2(b) and 2.2(c).

## 2.2.2   Control surfaces

In order to properly represent the control surfaces, two different grid approaches were tried.

**Control surface designed as an independent part**

A geometry with control surfaces meshed as independent parts with their own airfoil was implemented (Figure 2.3). Both a round (Figure 2.3(a)) and a pointy (Figure 2.3(b)) leading edge geometry were tried. The movement of the control surface is achieved with a rigid rotation of the whole part around the hinge. The main advantage of this approach is that a rotation of the control surface does not deform the grid itself. This is important to ensure that any variation on the coefficients is directly related to a movement of the control surface and not by changes in the mesh panel distribution.

After testing this approach with different airfoils it was found that the wake of the fixed part airfoil causes numerical instability if aligned with the chord of the control surface. This results in unexpected behaviours such as negative drag coefficients. For this reason it was decided to discard this approach.

**Control surface designed as part of the airfoil**

The alternative to the approach explained above is deforming the airfoil to represent the movement of the control surface. The airfoil mesh is cut on the control surface hinge vertical line, rotated rigidly (Figure 2.4), and re-joined to the fixed part (Figure 2.5).

The perturbation of a rotation of the control surface on the force and moment coefficients needs to only depend on the changes of the geometry itself and not on the local effects of the grid panels deformation. For this reason when the flap is rejoined to the airfoil the least possible number of panels are modified.

In order to minimize the deformations needed for the junction, the airfoil is meshed so that the hinge vertical line cuts the mesh exactly on a grid point, without cutting any panel. This is achieved simply by using the hinge arc-length coordinates as the boundary between the forward and aft mesh distributions.

After the control surface rigid rotation, the free edges on the flap are joined to the fixed part of the airfoil. If there is a gap, (like in the upper side of the airfoil in Figure 2.4), this is filled with an arch
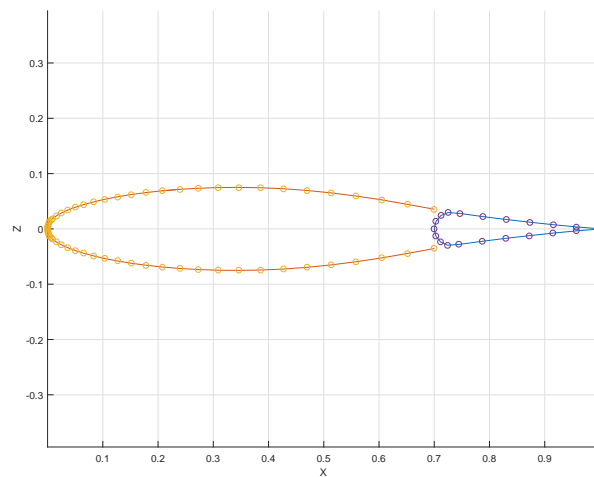
(a) Round leading edge
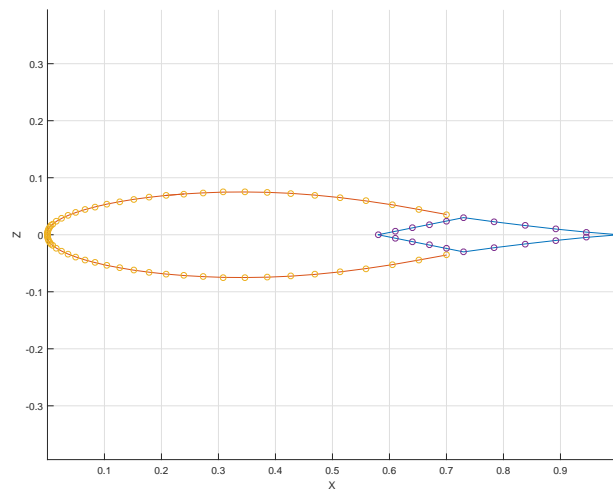


(b) Pointy leading edge

FIGURE 2.3: fixed wing and control surfaces modelled
as separate parts on FX-L-150 airfoil.

centred on the hinge point. If the two parts overlap (like in the
lower side of the airfoil in Figure 2.4) the control surface geometry is
shortened. The link geometry is made by deforming the control
surface mesh without changing the number of panels, by changing
the coordinates of only 4 grid points on the control surface mesh, 2
for the upper side and 2 for the lower side. This parameter (number
of points that may by repositioned for the control surface junction)
in the code can be changed, but the need to do so would only arise
for very thick meshes and/or very large deflection angles.
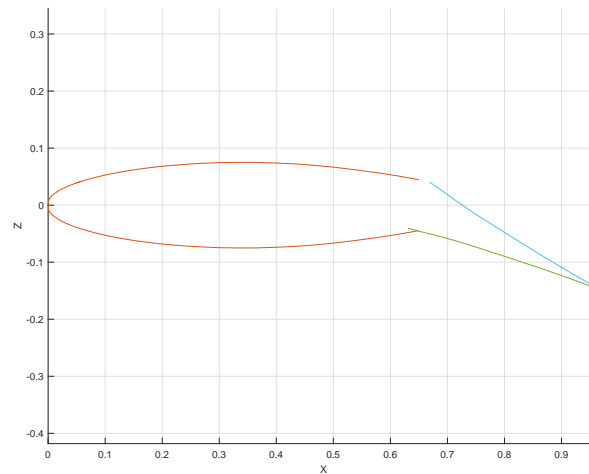
FIGURE 2.4: Flap geometry cropped and rotated rigidly.
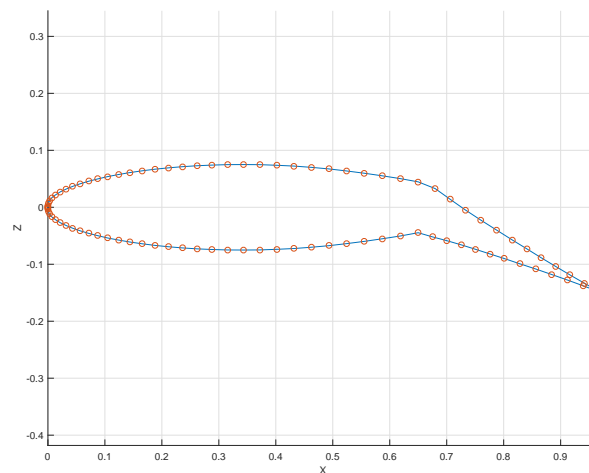


FIGURE 2.5: Mesh of the deformed airfoil.

The script that was implemented allows the creation of a control surface with any given hinge $x$ and $z$ coordinates, with the option to place the $z$ coordinate automatically on the airfoil outline. Note that the placement of the hinge outside of the airfoil geometry is possible and may be used to represent fowler-type flaps.

The airfoil subroutine inputs are:

- Airfoil coordinates file name.

- Number of panels.
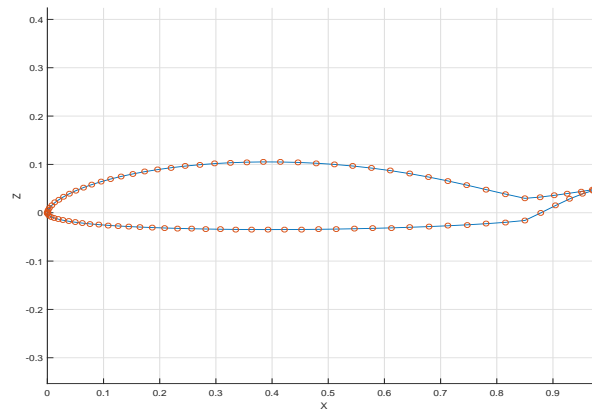
(a) Mesh of the airfoil DU04-140 with a flap at 85% of the chord deflected by $-20°$, hinge on the lower outline.
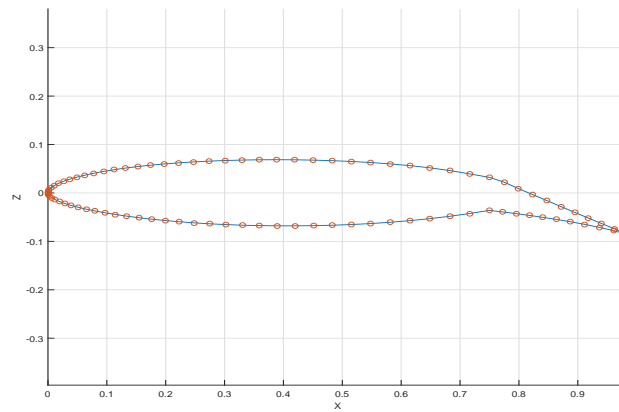


(b) Mesh of the airfoil DU86-137-25 with a flap at 75% of the chord deflected by $20°$, hinge on the mean chord.
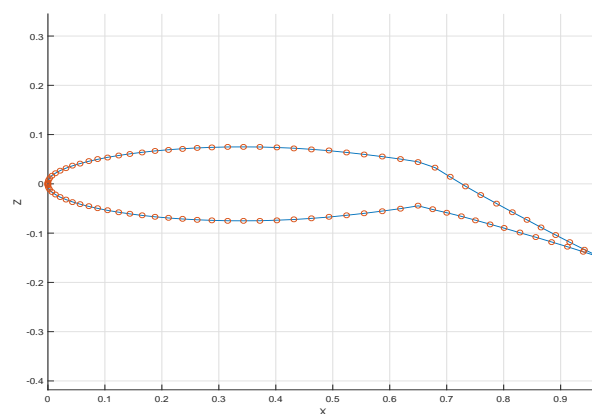


(c) Mesh of the airfoil FX-L-150 with a flap at 65% of the chord deflected by $25°$, hinge on the mean chord.

FIGURE 2.6: Control surfaces of the V5 "Rondone".

- Hinge axial coordinate, which is also is used to separate the forward and aft mesh distributions.

- Hinge vertical coordinate (optional, default value is 0).

- Control surface deflection angle (optional, default value is 0).

- $a_p$ parameter in aft panel distribution (optional, default value is 0).

The outputs are:

- The vectors $\{x\}$ and $\{z\}$, containing the coordinates of 2D airfoil mesh with unit chord

- The indices of the of the airfoil upper and lower hinge points (needed for the 3D mesh)

Figures 2.6(a), 2.6(b) and 2.6(c), represents the V5 wing, horizontal empennage and vertical empennage airfoils with deflected aileron, elevator and rudder respectively.

## 2.3 Lifting surface

The lifting surface subroutine is used to generate the wing and the empennages. It is able to mesh a lifting surface made of any given number of geometrical sections, each with an independent airfoil geometry, sweep, dihedral, taper ratio and pitch. It also generates twisted sections or airfoil transition sections with continuously changing airfoil shapes, and transforms a part of the body in a deflectable control surface.

The lifting surface subroutine discretizes the wing along the 25% chord line. The 2D airfoil grids are distributed along this line after having been generated with the Airfoil subroutine, scaled and rotated according to the wing geometry.

The inputs, for a finite wing made of $N$ sections and $M$ control surfaces are:

- The number of span-wise panels (longitudinal discretization).

- The number panels along the airfoil surface (sectional discretization).

- The $N + 1$ span-wise coordinates of the stations at the edges the $N$ sections that define the wing geometry. These can be optionally input either as absolute $y$ coordinates or as arc-length coordinates along the 25% chord line (mandatory for designing wings with winglets)

- The $N + 1$ airfoil file names at each station.

- The $N + 1$ chord lengths at each station.

- The $N + 1$ pitch angles at each station.

- The $N$ dihedral angles at each section.

- The $N$ 25% chord sweep angles at each section. Optionally, for wings with a straight leading or trailing edge (such as the V5 wing), the user may only specify one sweep angle for that edge, and which edge to keep straight. All of the subsequent sweep angles will be computed accordingly.

- The $2M$ span-wise coordinates of the stations delimiting the control surfaces.

- The $2M$ coordinates of the airfoil hinge position (as % of the chord) at the edges of each control surface. In case the two hinges have different coordinates the hinge line will vary linearly along the span.

Each geometrical input needs to be entered as a vector where the values on the fuselage side are first and the ones towards the tip are last.

The outputs are:

- The $[x]$,$[y]$ and $[z]$ matrices containing the coordinates lifting body points, organized in such a way that each row is an airfoil grid

### 2.3.1 Span-wise coordinates distribution

The span-wise coordinates of the 2D airfoils will determine the span-wise position and dimension of the panels along the 25% chord line.

In order to reduce the discretization error on the pressure coefficient it is important to refine the grid where a higher span-wise pressure gradient is to be expected. This happens near the tips, at the edges of the control surface and along a twisted section. For this purpose, to keep the code suitable for any given lifting body mesh and have its discretization defined by only one longitudinal and one sectional parameter (number of panels), a function is designed to generate a suitable coordinate distribution with any given number of local mesh thickening centres. Each thickening centre is defined by its span-wise position and strength, which are parameters of the distribution function.

The following function is used as a base:

$$g_0(x|a, b) = \frac{\cos(b) - \cos(ax + b)}{\cos(b) - \cos(a + b)}, \tag{2.3}$$

with $a = \pi$ and $b = 0$ the thickening is symmetric, as shown in Figure 2.7.

In order to independently control the left and right thickenings, $a$ and $b$ are redefined as functions of the left and right weight parameters $p_l$ and $p_r$:

$$a = \frac{\pi}{2}(p_l + p_r) \tag{2.4a}$$

$$b = \frac{\pi}{2}(1 - p_l). \tag{2.4b}$$

By substituting them into Equation 2.3 the distribution function is redefined:

$$f_0(x|p_l, p_r) = g_0(x|a(p_l, p_r), b(p_l, p_r)). \tag{2.5}$$

Figure 2.8 shows how $p_l$ and $p_r$ influence the distribution of Equation 2.5.

In order to have $n$ centres of increased thickness along a wing span, defined by the vectors containing their coordinates $\{\xi\}$ and weights $\{p\}$, a new distribution function is created by combining multiple piecewise sub-functions $f_0(x|p_i, p_{i+1})$ from Equation 2.5:
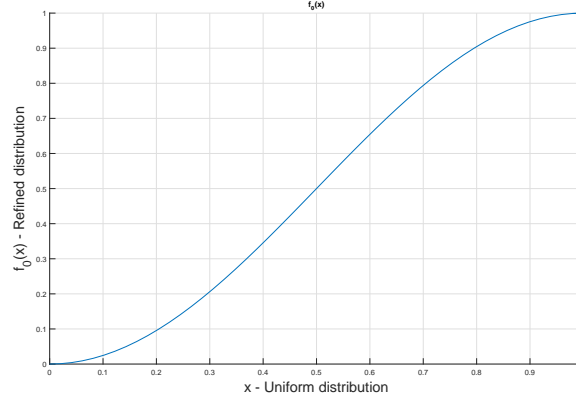
(a) Graph of $g_0(x|\pi, 0), x \in [0, 1]$ .
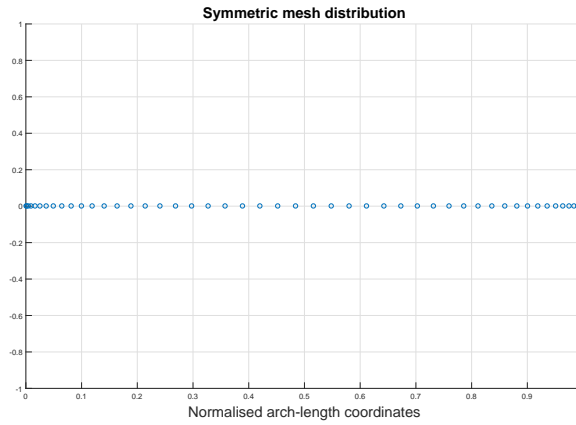


(b) Refined mesh distribution (50 points).

FIGURE 2.7: Distribution function $g_0(a|a, b)$ with $a = \pi$ and $b = 0$.

$$f\left(x|\{p\}, \{\xi\}\right) = \Sigma_{i=1}^{n-1}\text{rect}(x|\xi_i, \xi_{i+1}) \left( (\xi_{i+1} - \xi_i) \, f_0 \left( \frac{x - \xi_i}{\xi_{i+1} - \xi_i} \middle| p_i, p_{i+1} \right) + \xi_i \right),$$

$$(2.6)$$

where $\text{rect}(x|\xi_i, \xi_{i+1})$ is the rectangular pulse function between points $\xi_i$ and $\xi_{i+1}$. Figure 2.9 presents an example of the resulting distribution from Equation 2.5 with two centres of increased thickness positioned along the span in addition to the two at the edges.

Since the pressure gradient at the wing tips is expected to be high, the thickness provided by Equation 2.6 at maximum value of the weights $p_1 = 1$ and $p_n = 1$ is deemed not enough. A simple way to further increase the mesh refinement at the boundaries is to use the output of the simple distribution function $f_0(x|p_l, p_r)$ (Equation 2.5)
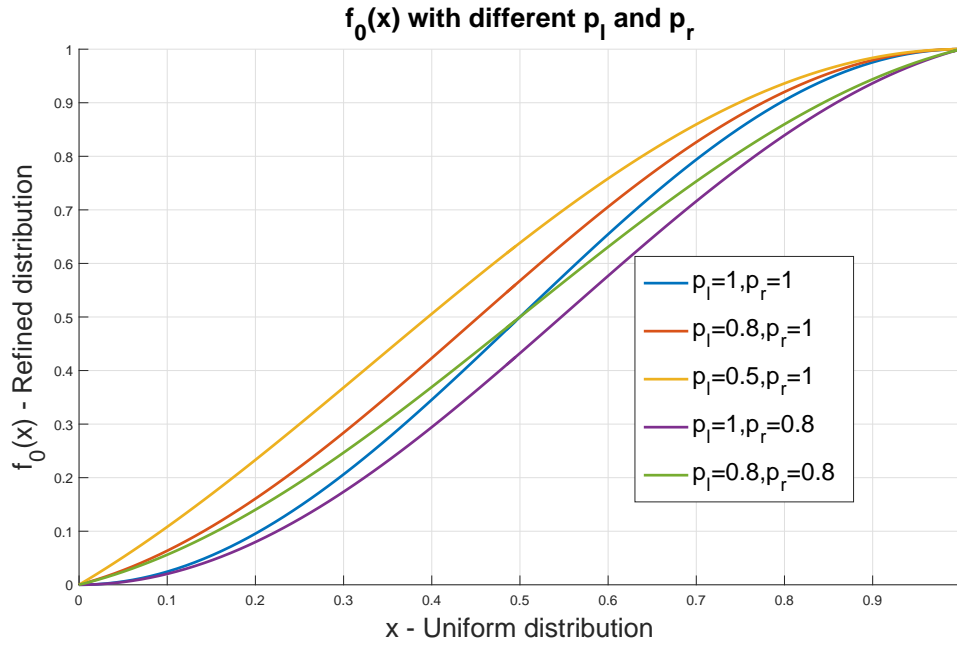
FIGURE 2.8: $f_0(x|p_l, p_r)$ span-wise distribution function with different $p_l$ and $p_r$ values.

as input of the main distribution function $f\left(x|\{p\}, \{\xi\}\right)$ (Equation 2.6) instead of a uniform distribution:

$$\overline{f}\left(x|\{p\}, \{\xi\}, p_l, p_r\right) = f\left(f_0(x|p_l, p_r)|\{p\}, \{\xi\}\right). \tag{2.7}$$

In this function both $(p_l, p_r)$ and $(p_1, p_n)$ work for and increased thickness at the two ends. The difference between them is that first couple increases the local thickness by redistributing the points on the whole mesh, while the second only redistributes the points on the wing tip sections.

The last thing needed in order to automatically mesh the lifting surface is to set values of $p_i$, $p_l$ and $p_r$ that work for any given lifting body geometry. The goal is to achieve a panel span-wise dimensions distribution that is inverse proportional to the anticipated pressure gradient.

The values $p_i$ are set as follows:

- $p_i = 0.8$ at the edges of a control surface.

- $p_i = 0.7$ at the edges of a twisted section.
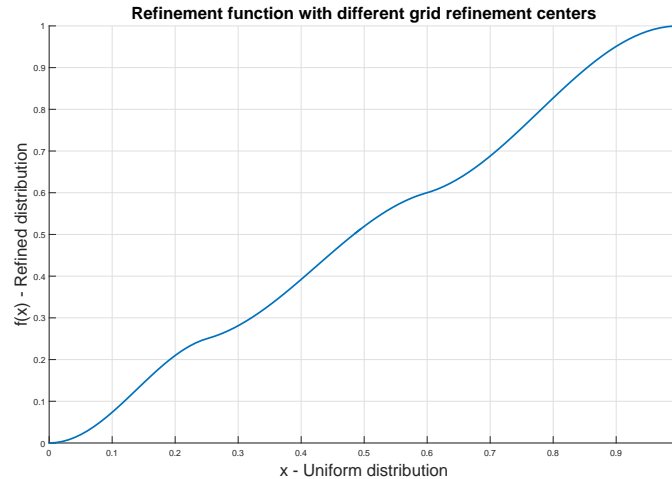
(a) Graph of $f(x|\{\xi\}, \{p\}), x \in [0, 1]$.
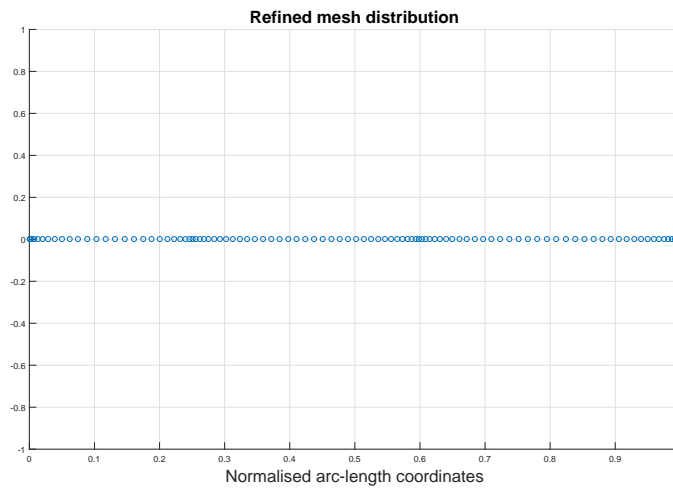


(b) Resulting span-wise points distribution (100 points).

FIGURE 2.9: Distribution function $f(x, |\{\xi\}, \{p\})$ with $\{\xi\} = [0, 0.25, 0.6, 1]^T$ and $\{p\} = [1, 0.8, 0.8, 1]^T$.

- $p_1 = 0.5$, $p_l = 0.8$ at the lifting surface edge on the fuselage side.

- $p_n = 0.5, p_r = 1$ at the lifting surface edge on the wing tip.

The final V5 "Rondone" wing distribution function is represented in Figures 2.10 and 2.11.

When transforming the output of the distribution functions to an actual grid of arc-length coordinates to be used for the lifting surface mesh, nothing ensures that an airfoil grid will be placed at the exact boundary station between two wing sections. For a better representation of the wing outline and to ensure that the geometry does not chance with changes in the span-wise number of panels,
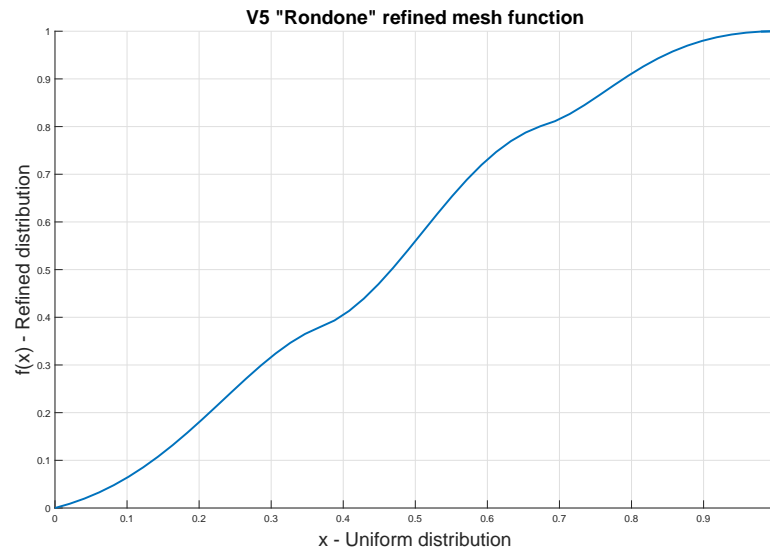
FIGURE 2.10: Panels distribution function for the V5 "Rondone": $\{\xi\} = [0, 0.3810, 0.8038, 1]^T$.
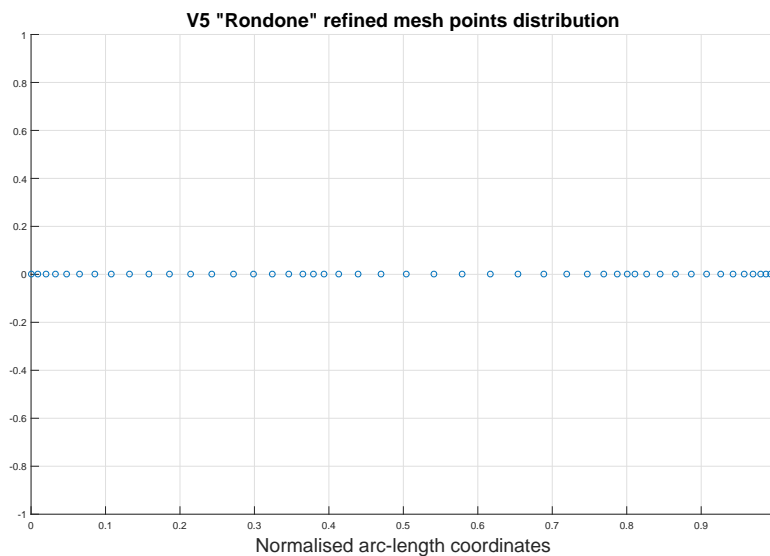


FIGURE 2.11: Resulting span-wise normalized panels distribution for the V5 wing with 50 points.

the grid airfoils distribution in each section needs to be slightly scaled in such a way that an airfoil grid is placed exactly at the boundary station

This is achieved with a (small) uniform expansion or compression of the local distributions of the two sections on the side of each station, aimed at moving the point that was already closest to the boundary coordinate exactly on top of it. In order not to move the boundary point at the other edge of the two sections, the left distribution is scaled with respect to its left edge, while the right one

is scaled with respect to its right edge (Figure 2.12).
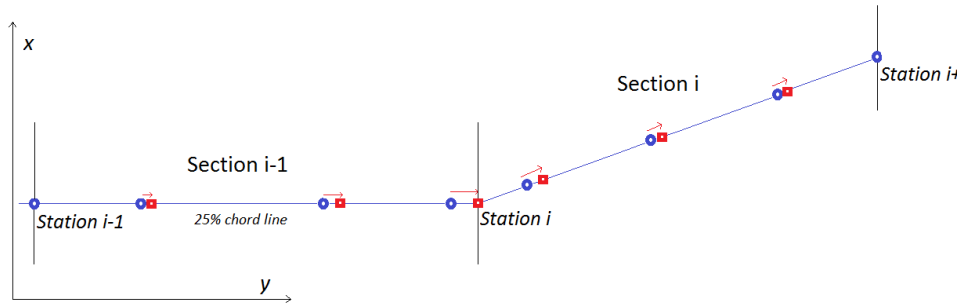


FIGURE 2.12: Scaling of the grid points on sections $i-1$ and $i$ to place a grid point on top of station $i$ (top view). The starting points are blue circles while the final ones are red squeres.

### 2.3.2   3D lifting body generation

The mesh of the lifting surface is generated by placing 2D airfoil grids along the longitudinal axis, which is the 25% chord line of the lifting body, on the arc-length coordinates given by the span-wise distribution.

The lifting body is first built on the $x - y$ plane (meaning that the $25\%$ chord line will remain on this plane), where the $x$ axis is aligned with the asymptotic air velocity when the aerodynamic angles and the pre-pitch are null and $y$ is the axis along the span. Dihedral angles are added later.

**Planar geometry**

The 2D airfoil grids at the input stations are generated first, all lying in a $x - z$ parallel plane and all with the same number of grid points. The airfoils shape, pitch, chord, and position along the $x$ axis (dependent on the sweep angle) at each station defines the wing geometry.

The airfoils positioned at the stations generates the lifting body geometry with a linear interpolation of their coordinates along the span. The airfoil points are interpolated on the span-wise

coordinates given by the distribution function $\overline{f_0}$ corrected to fit in the lifting body geometry.

This method ensures that the wing proprieties changes linearly along the section. If more than one type of airfoil is used for the wing, there will be a transition section between the two different airfoils. Similarly, a section between two stations with different pitch angles will be twisted gradually.

**Control surfaces**

The control surfaces are the only feature that requires a discontinuity in the planar geometry between two sections. In order to represent this, each control surface introduced in the wing splits it in different blocks. The total number of blocks in a lifting surface is thus $2M + 1$, where $M$ is the number of control surfaces (Figure )

For each station at the edge of a control surface, two airfoil grids are created, one features the deflected control surface geometry, used for the interpolation of the airfoil grids on the aileron side, the other one features the non-deformed geometry and is used for the interpolation of the airfoil grids on the side without the control surface.

In order to avoid interference problems between the deflected surface and the non deflected panels on its sides, the grid points at the edges of the control surface are slightly moved along the longitudinal towards the interior of the flap geometry.

**Dihedral angles**

The dihedral angles are introduced with a series of rigid rotations.

First, each station is assigned the value of the relative dihedral angle between the two bordering sections (the difference between their dihedral angles), except for the first station, which is assigned the full value of the dihedral of the first section, and the last station, which is not assigned any value.
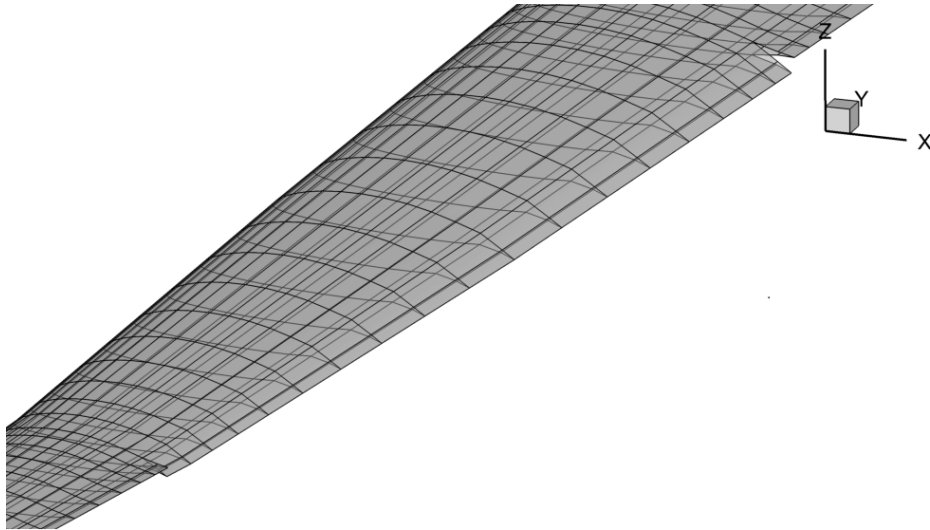
FIGURE 2.13: V5 "Rondone" aileron deflected by $20°$.

The rigid rotations are introduced from the wing tip section, which is rotated by the relative dihedral value of the station on its non-free end. The axis of rotation is aligned with the $x$ axis and passes from the 25% chord point of the airfoil of the station. After that, all the other relative dihedral rotations are enforced in order from the tip to the root. Each one prompts a rigid rotation by its relative dihedral value of all the sections between its station and the free end. In other words, each section will undergo a rigid rotation for any non-zero relative dihedral value between itself and the wing root, each rotation will be centred on a different axis.

In order for the wing sections to connect neatly after the dihedral rotation, the airfoil grids on the input stations will undergo a rotation of half of their relative dihedral angle, and will be scaled to meet the projections of the geometry of both sections (Figure 2.14)
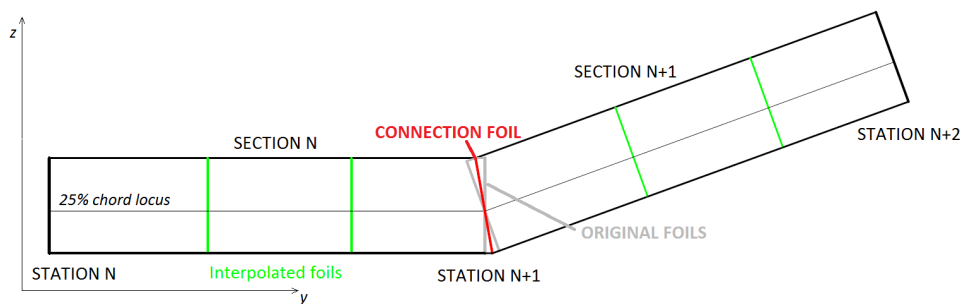


FIGURE 2.14: Front view (leading edge) of the connection between two sections with $20°$ relative dihedral.

Thanks to the dihedral code structured in this way, the lifting surface wing mesh is able to generate the meshes of wing sections with high relative dihedral angles without deformations in the airfoils. This is especially useful in case the user is interested in including winglets in the analysis.

Figure 2.15 represent the final mesh for the V5 "Rondone" wing.

## 2.4   2D fuselage geometry

The 2D fuselage geometry subroutine is the base unit of the Fuselage subroutine. It generates the geometry of the fuselage transversal section given the following inputs:

- The local horizontal diameter $D_H$.

- The local vertical diameter $D_V$.

- The local vertical offset from the $x$ axis $H_0$.

Given the infinite possibilities of fuselage geometries, this subroutine is dedicated to the V5 "Rondone" exact geometry. However the fuselage section it generates is a good approximation of the one found in most gliders. Even if that was not the case, the effect on the end results may still be limited considering the small influence of non lifting bodies on the overall force and moment coefficients.

The basic geometry is made by two semicircles joined by two straight segments so that the total height and width match the input diameters:

- If the horizontal diameter is higher than the vertical one, the two semicircles of diameter $D_V$ are joined horizontally by segments of length $D_H - D_V$ as in Figure 2.16(a).

- If the vertical diameter is higher then the two semicircles of diameter $D_H$ are joined vertically by two segments of length $D_V - D_H$ as in Figure 2.16(c).

- If the two dimensions are equal than the sectional geometry is circular of diameter $D_V = D_H$ as in Figure 2.16(b).
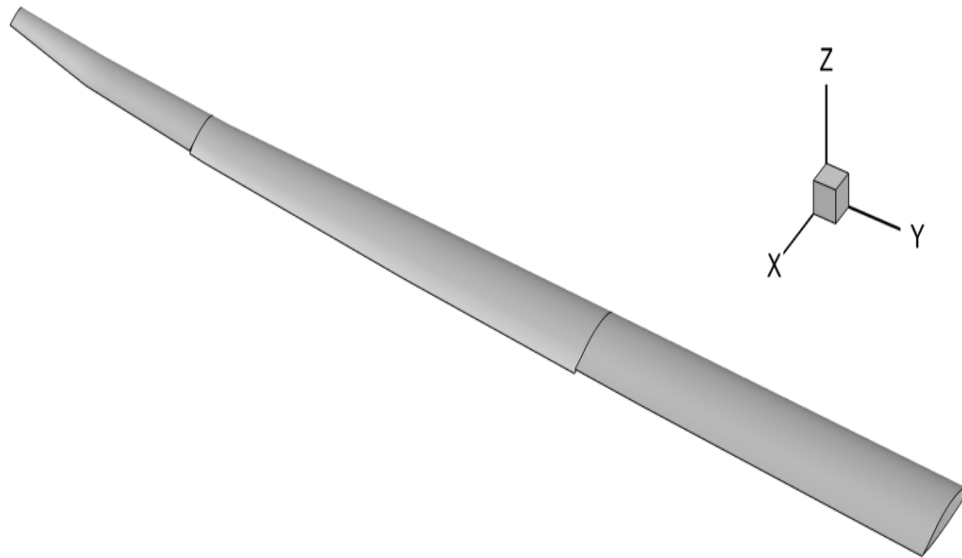
(a) Wing blocks.
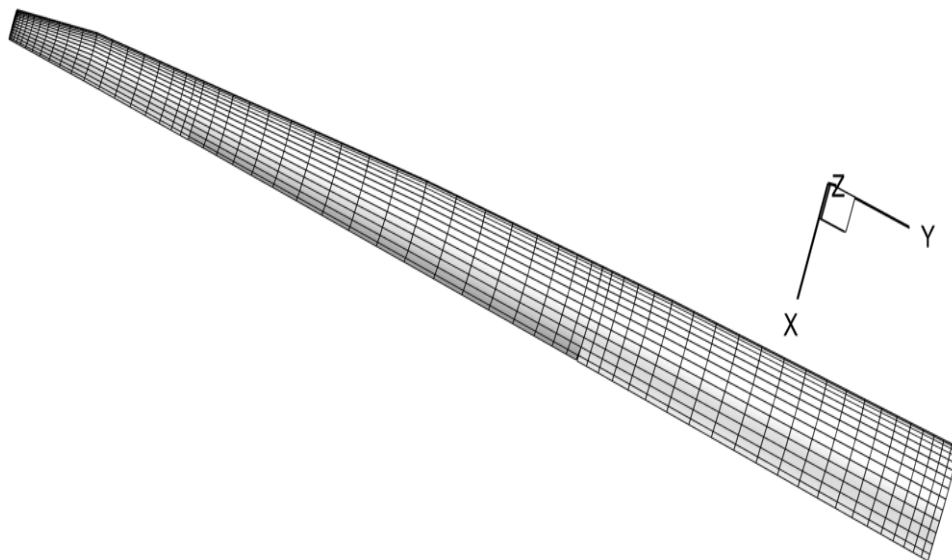


(b) Wing mesh, top view.
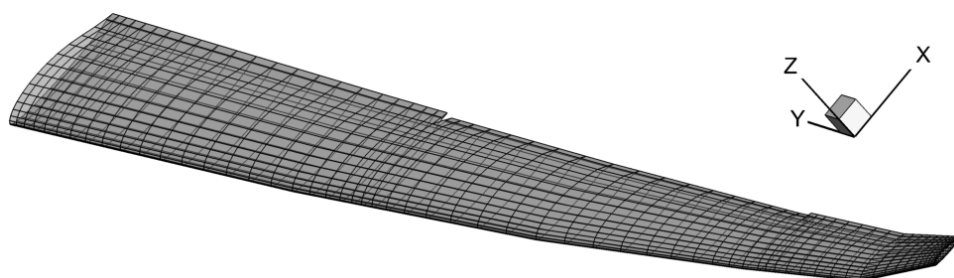


(c) Wing mesh, leading edge view.

FIGURE 2.15: V5 "Rondone" left wing mesh, with 30 airfoil panels, 40 span-wise panels and 20° aileron deflection.
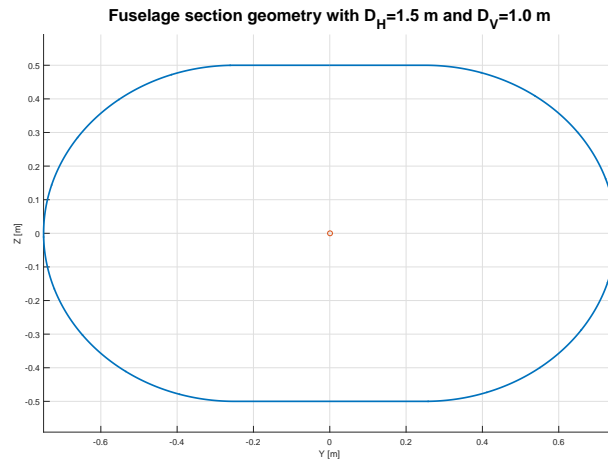
(a) $D_H = 1.5, D_V = 1$.
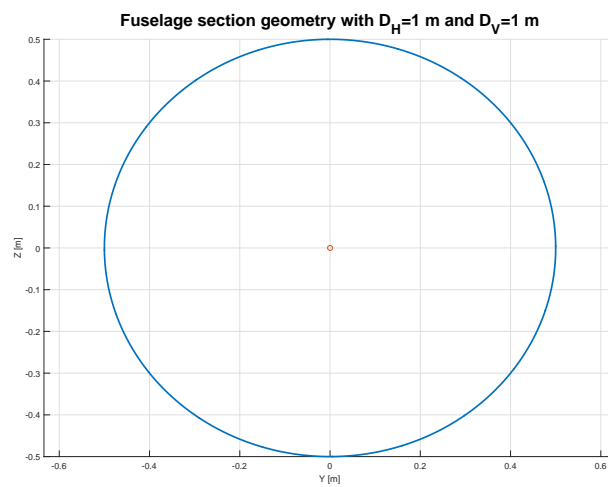


(b) $D_H = 1, D_V = 1$.
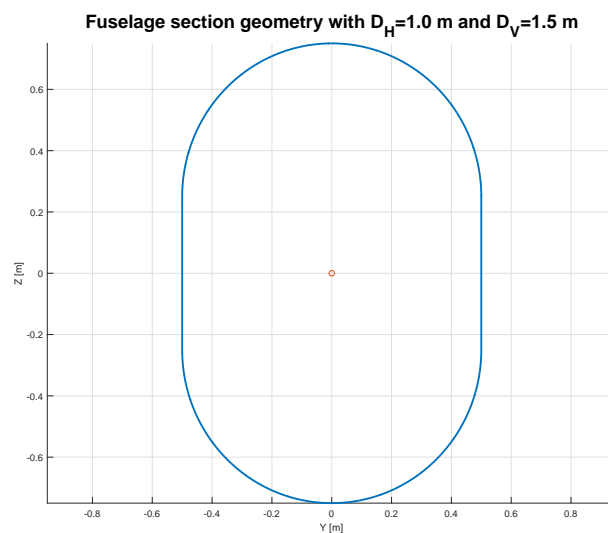


(c) $D_H = 1, D_V = 1$.

FIGURE 2.16: V5 "Rondone" section geometry with different $D_H$ and $D_V$ values.

After applying the $H_0$ offset to the $z$ coordinates, the subroutine computes the end geometry as a curve row of points in the $y - z$ plane that describes the half section, symmetric with respect to the $x - z$ plane.

The outputs are the vectors of the point coordinates $\{y\}$ and $\{z\}$ as well as the the vector of arc-length coordinates $\{s\}$. The mesh describing this geometry is thick (200 points) because it will need to be interpolated later to create the 3D mesh.

## 2.5 Fuselage

The Fuselage subroutine is tasked with the generation of the full fuselage mesh. The symmetry of the fuselage with respect to the $x - z$ plane is exploited through the whole process. The inputs of the subroutine are:

- A set N coordinates for along the $x$ axis for the definition of the geometry.

- The sets of correspondent N horizontal diameters, vertical diameters and offset coordinates.

- The wing mesh root airfoil grid

- The vertical empennage root airfoil grid

The subroutine generates a mesh representing the fuselage geometry and provides a point-by-point connection area for the wing and vertical empennage. This is a precaution to avoid possible numerical instability due to singularities in the sources and vortices distribution in COMPA, or the establishment of non physical flows due to holes in the geometry.

First, the longitudinal discretization of the fuselage mesh needs to be established. In particular, in the areas where the wing and the vertical empennage link with the fuselage the discretization along the $x$ axis must be the same as that of the connected lifting body.

In order to keep the panel size regular the grid distribution in the wing connection area is used, mirrored and properly scaled, for the front section of the fuselage as well. The same applies to the panel

distribution for the vertical empennage connection area which is used for the tail boom, while the fuselage section between the wing trailing edge and the tail boom is filled with panels of proportional size to that of last panels of the trailing edge at the wing root. Figure 2.17 shows the longitudinal discretizaion of the V5 fuselage.

The horizontal and vertical diameters and the offsets are computed at the longitudinal grid coordinates with a spline interpolation of the input values (Figure 2.17). For each longitudinal coordinate the geometry of the transversal section is then created with the dedicated subroutine. The panels distribution in the transversal section is uniform and is obtained by interpolating the geometry with a uniformly spaced arc-length vector. In order to obtain panels with proportionate dimensions, the number of panels in the transversal section is tied to the longitudinal discretization . This method generates a fuselage mesh where all the discretization parameters are a function of the wing and vertical empennage discretization. Those needs to be chosen with some testing of their effects on the aerodynamic coefficients.

In the wing region the mesh needs to have holes for the wing to connect point-by-point. In order to do this the lower and upper grid points of the wing root airfoil are projected on the fuselage contour along the $y$ axis. The fuselage sections can then be trimmed at the exact junction points. The easiest way to create the wing root hole in the fuselage is to break the fuselage into different blocks: front (forward from the leading edge), upper wing region (connects the upper sides of the wing root airfoil meshes), lower wing region (connects the lower sides of the wing root airfoil meshes) and back (aft of the wing trailing edge), which may be further split if needed. In particular, the mesh of the lower and upper wing region blocks is created by computing the arc-length coordinates of the wing-fuselage connection points for each section, so that the grid points can be interpolated from the geometry with a suitable arc-length distribution. The fuselage-wing connection area is shown in figure 2.18.

Similarly to the wing-fuselage connection, the vertical empennage-fuselage connection is made by projecting on the fuselage the root airfoil grid points of the empennage along the $z$

(a) $x - z$ plane.

(b) $x - y$ plane.
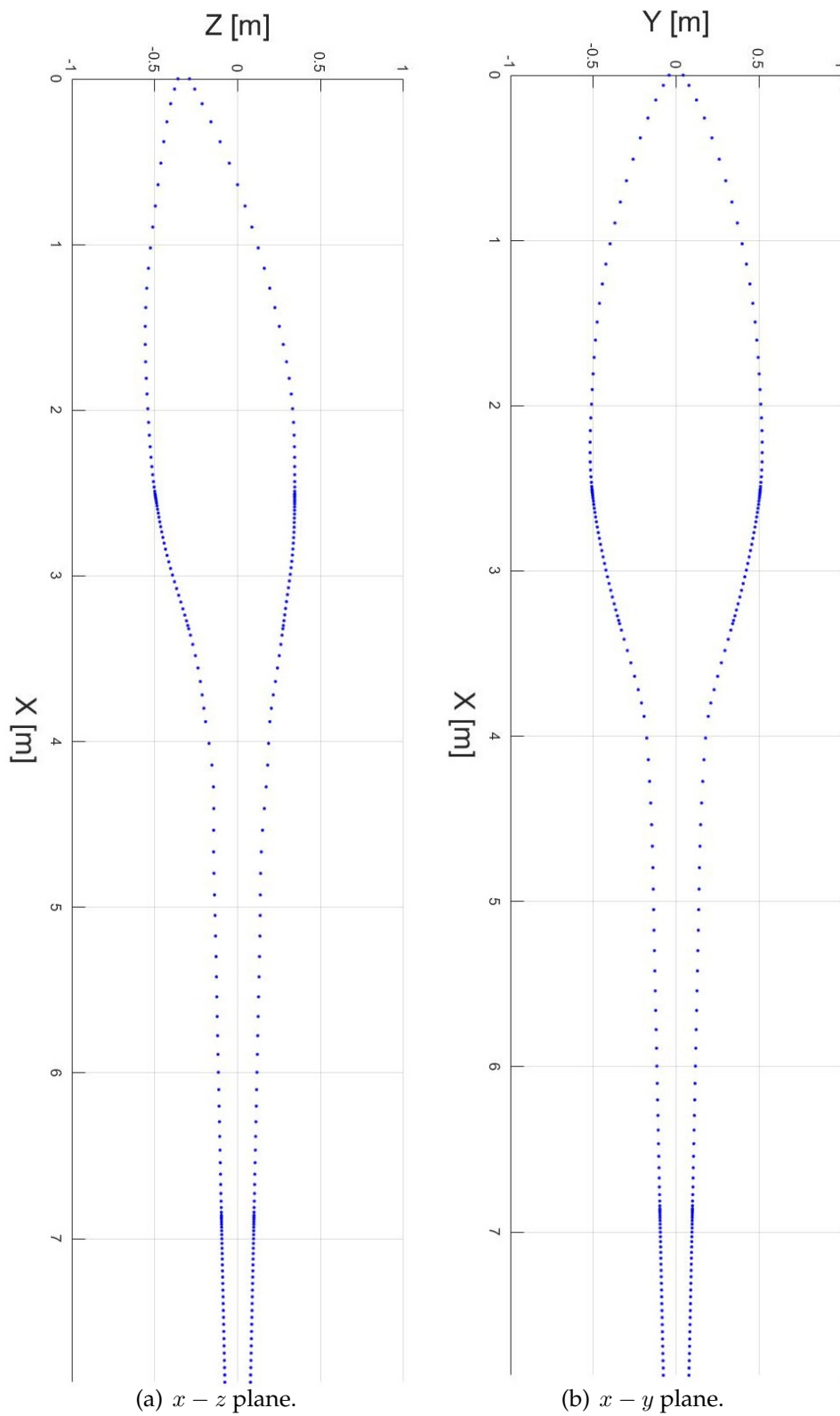
FIGURE 2.17: Longitudinal grid discretization and spline interpolation of the sectional proprieties of the fuselage.
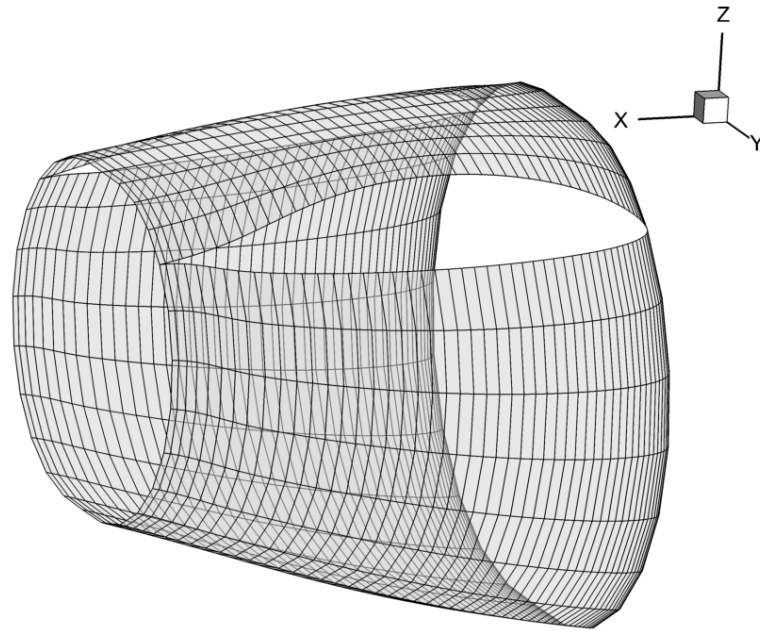
FIGURE 2.18: Fuselage-wings connection mesh.

axis. The hole for the tail connection is made by computing the arc-length coordinates of the connection points so that the panels may only be placed on the external side. This approach for the fuselage-vertical empennage connection requires that the fuselage is always wider or equal to the empennage airfoil thickness. If that stops being the case at some point of the empennage chord, the code will keep generating semicircular fuselage sections of diameter equal to the fin root local thickness until the trailing edge. The last part of the fuselage vertical empennage junction stops being symmetric on the $x - z$ plane due to the possible deflection of the rudder. The fuselage-empennage connection area is shown in figure 2.19.
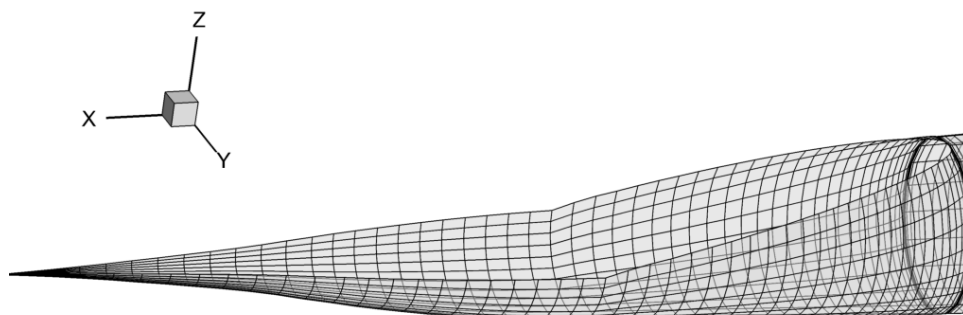


FIGURE 2.19: Fuselage-vertical fin connection mesh.

In this code the fuselage mesh is made by six blocks: front, upper wing region, lower wing region, tail cone, tail boom and fuselage-fin connection area. The full fuselage geometry is shown in Figure 2.20.

## 2.6 Junctions

The Junctions subroutine takes care of creating clean, point-by-point links between the various parts that needs to be connected. All of those links involve the connection of a lifting body to a curved surface so, in order to make the connection, a deformation of the grid points along the longitudinal axis of the lifting surface is needed. This will cause the airfoil meshes to develop in the 3D space and lose their planar geometry.

Extensive testing with COMPA has shown that deforming the airfoil grids along the longitudinal axis of the lifting body is feasible only if the deformation is symmetric with respect to the airfoil chord. An asymmetric deformation has been shown to always cause numerical instabilities in the flow, resulting in non-physical solutions with very high local pressures and velocities.

### 2.6.1 Wing-fuselage junction

The wing needs to be modified in order to meet the fuselage point-by-point. The code is implemented so that if the existing wing root happens to be inside the fuselage geometry its root panels will be deformed, if the wing root does not reach the fuselage an additional bock will be added. In any case the wing cannot be directly connected to the fuselage as it would require the asymmetric deformation of an airfoil on its longitudinal axis. For this reason and additional non lifting block is added to connect the wing edge to the hole in the fuselage.

The number of span-wise airfoil grids in the added block is computed in order to be keep the span-wise dimension of the panels similar to that of the wing root. Figure 2.21 offers a detailed view of the wing-fuselage connection
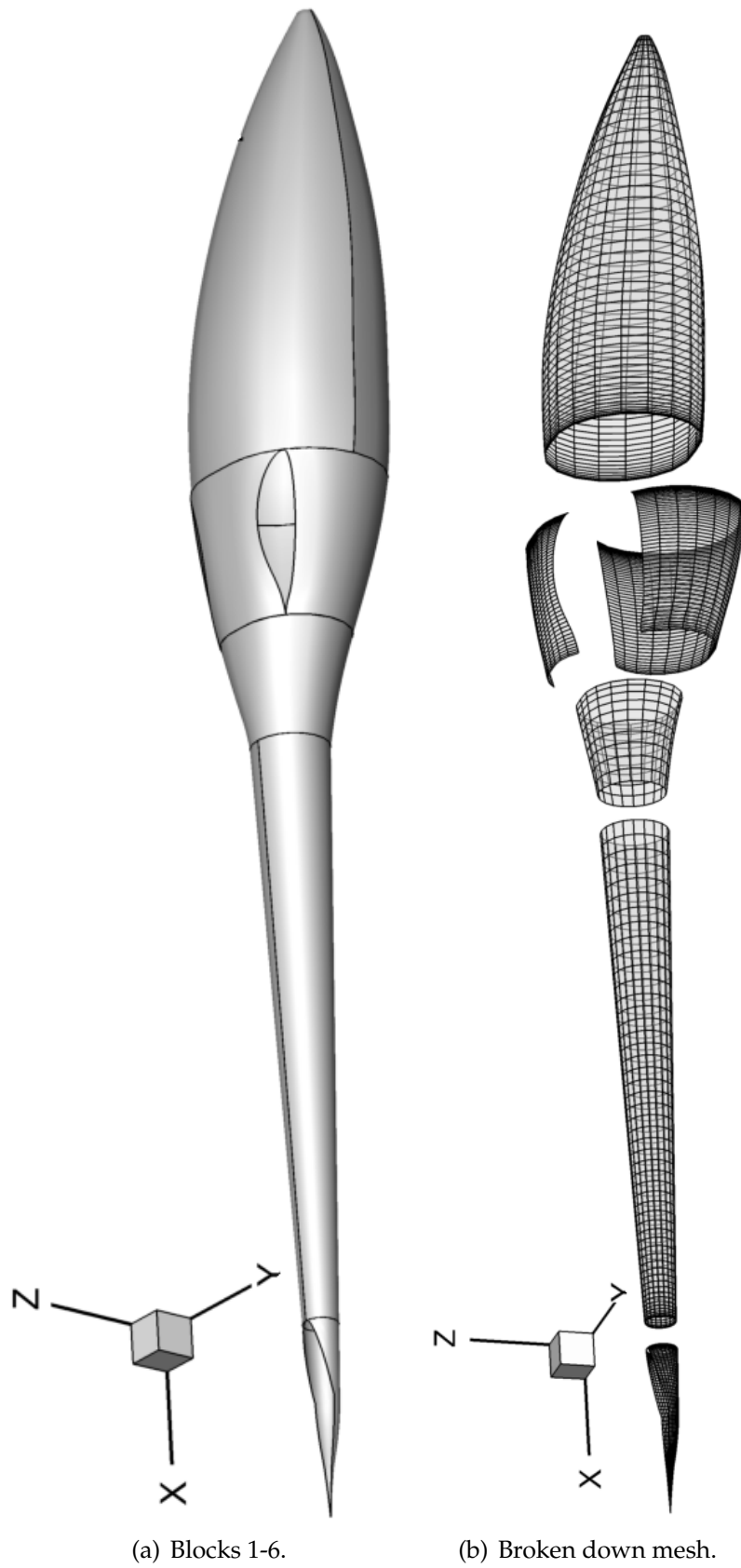
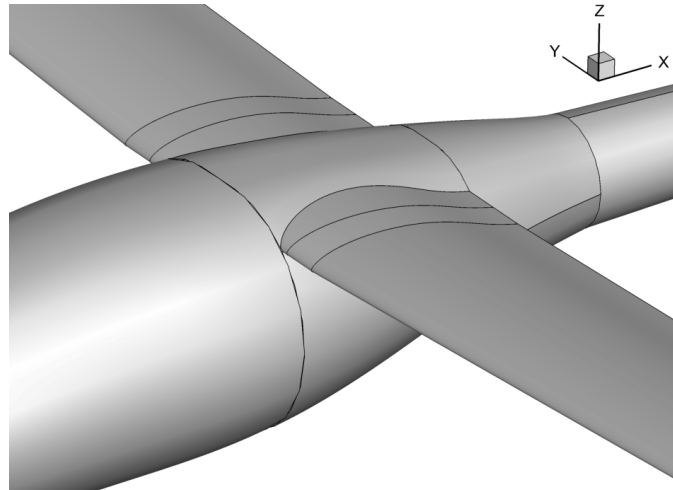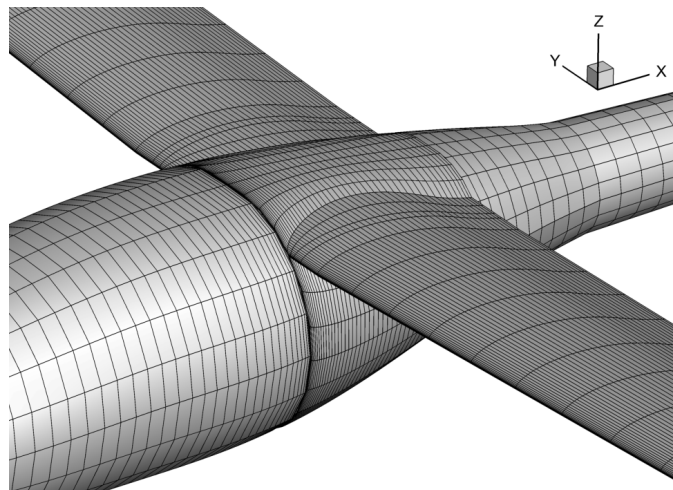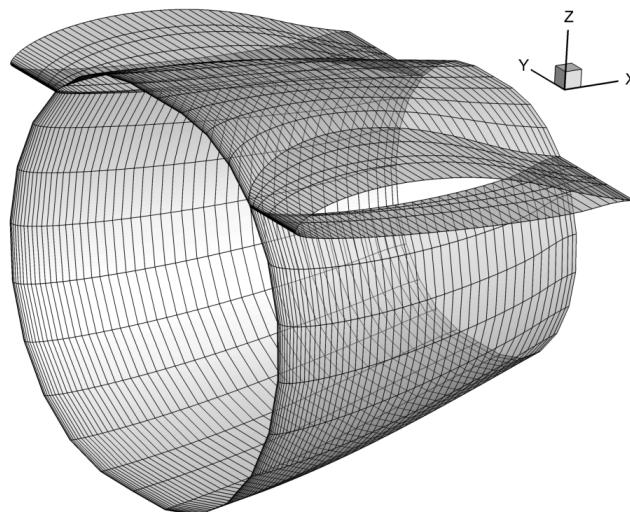(a) Blocks 1-6.    (b) Broken down mesh.

FIGURE 2.20: V5 "Rondone" fuselage mesh.

(a) Added blocks to the wing root.



(b) Added panels to the wing root.



(c) Detailed view.

FIGURE 2.21: Wing-fuselage connection.

## 2.6.2 Empennages junction

The junction between the horizontal and vertical empennages is the only connection between two lifting bodies in the geometry. The two bodies need to meet point-by-point for the length of tailplane fixed part, while the moving part (the elevator) needs to remain free. The code for this junction only generates half of it, taking advantage of the symmetry with respect to the $x - z$ plane

The tail-plane is modified first, its root grid points are stretched along the $y$ axis to match the shape of the vertical airfoil on top of the vertical fin (Figure 2.22(a)). This is done by interpolating half of the airfoil geometry on the $x - y$ plane at the $x$ coordinates of the root airfoil of the tail-plane. The tailplane root airfoil is then deformed along the $y$ axis to meet the interpolated coordinates, while its control surface is kept straight. The deformation is then spread to the outer airfoil grids for 10% of the tailplane span in order for it to be absorbed smoothly.

The vertical fin needs to join the tail-plane root airfoils point-by-point. In order to do this the top block of the vertical fin mesh needs to be heavily modified (Figure 2.22(b)): its top airfoil grid is swapped with one interpolated on the $x - y$ plane at the $x$ coordinates of the tail-plane root and deformed on the $z$ axis in order to meet its fixed part point-by-point. For the aft part of the airfoil, not joining the tail-plane, a uniform spacing similar to that of the last tailplane root panel is used. The other grid panels of the top block are the result of an interpolation between the unchanged airfoil mesh at the bottom of the block and the top airfoil mesh described above.

In order to complete the geometry, a non lifting patch is made to plug the hole left on the top of the tail-plane (Figure 2.22(c)).

## 2.6.3 Empennage-fuselage junction

Given that the fuselage is already meshed with grid points placed at the exact vertical empennage root airfoil $x$ and $y$ coordinates, all that is left to do is to deform the empennage root airfoil on the z axis in order to meet the fuselage point-by-point (Figure 2.23)
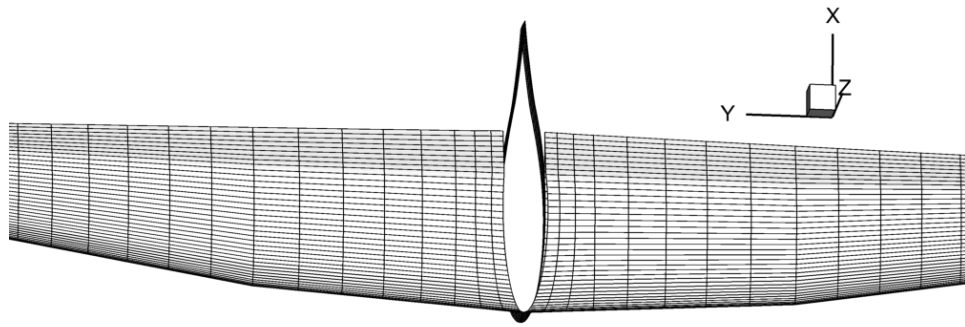
(a) Tailplane deformation.
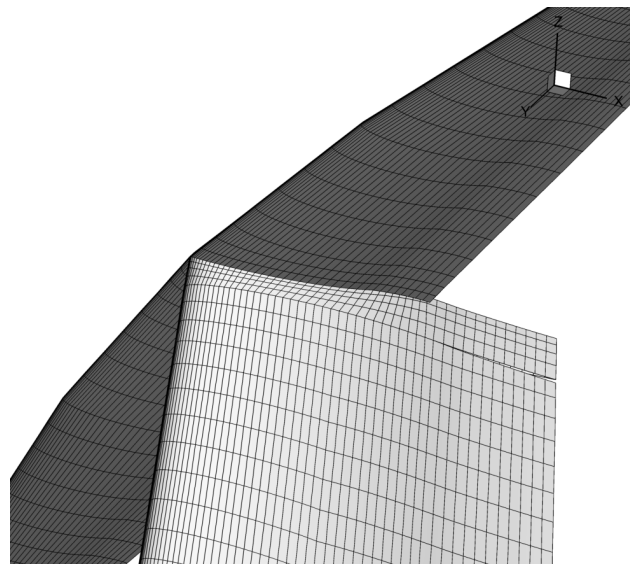


(b) Vertical fin deformation.
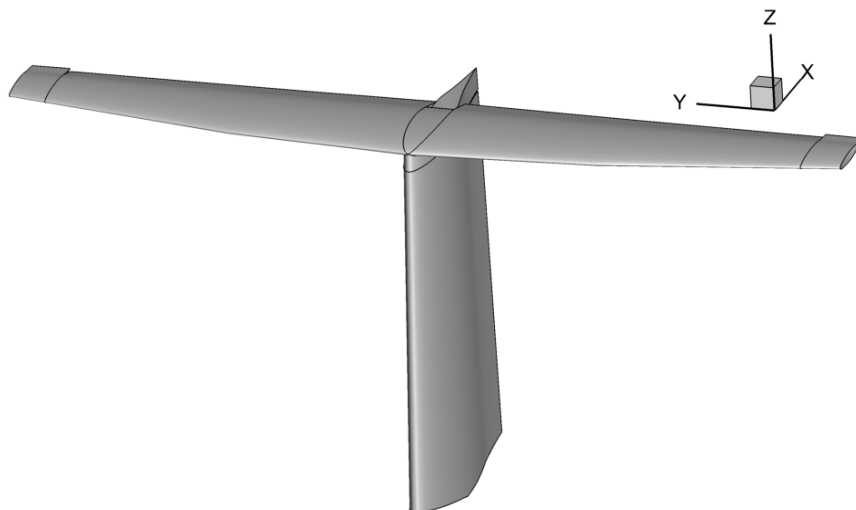


(c) Empennage assembly.

FIGURE 2.22: Horizontal-vertical empennages con-
nection.
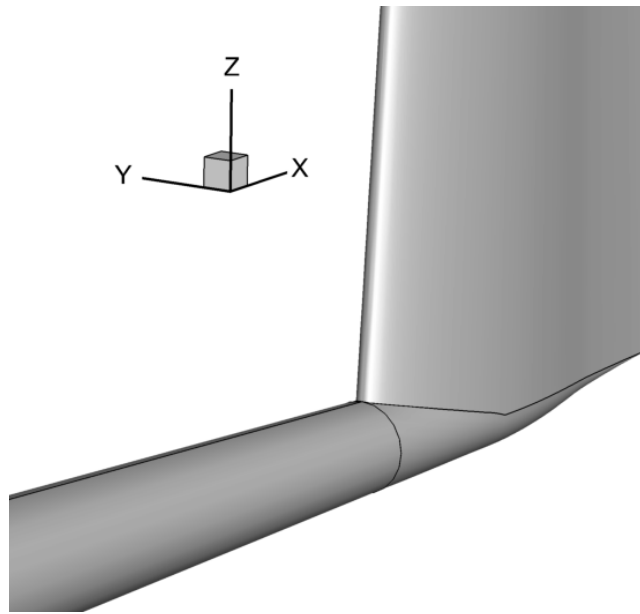
(a) Blocks.
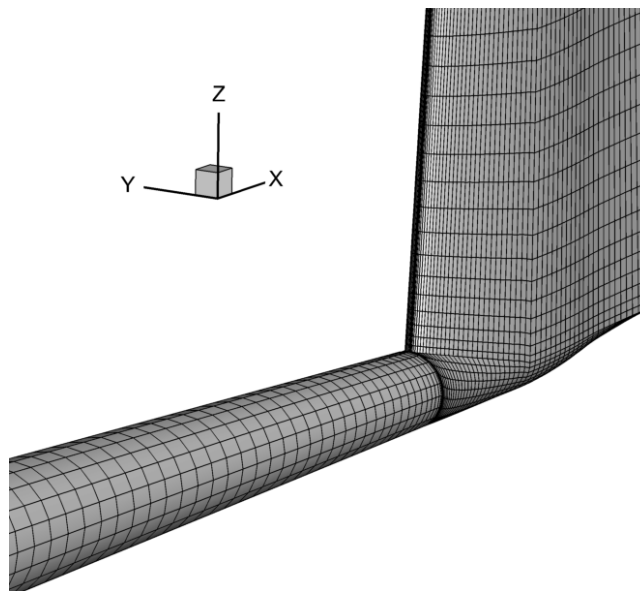


(b) Panels.

FIGURE 2.23: Vertical empennage-fuselage connection.

## 2.7 Aircraft mesh

The main "Aircraft mesh" routine generates the various parts using the existing subroutines and positions them in the right places. As required by COMPA, the mesh is built in a right-handed coordinate system with the $x$ axis pointing in the same direction of the asymptotic flow velocity at $0°$ AoA and the $z$ axis pointing up.

The required inputs are:

- The longitudinal and sectional number of panels for the wing and the empennages. The vertical empennage only requires a longitudinal (span-wise) number of panels since its airfoil grid refinement is related to that of the tail-plane.

- The fuselage, wing, and empennages geometry, in the format required by the fuselage and lifting surfaces subroutines. The vertical fin span is not required as it will be calculated in order to match the tailplane on one side and the fuselage on the other.

- The wing and horizontal empennage $x$ and $z$ offset coordinates and pre-pitch. The vertical empennage does not require those because it will be placed between the tailplane and the fuselage automatically.

- The angles of attack (AoA) and angles of side-slip (AoS) at which the analysis will be performed.

- The control angles.

The code will build the plane using the $x$ axis as the longitudinal axis of the fuselage, with the origin marking the nose of the sailplane and the axis pointing towards the tail.

Those are the steps for the full aircraft mesh generation:

- Wings mesh generation (Lifting surface subroutine) and positioning. Both wings need to be generated separately because the aileron deflection makes them asymmetric.

- Horizontal empennage mesh generation (Lifting surface subroutine) and positioning. The $x - z$ plane symmetry is exploited.

- Vertical empennage mesh generation and positioning.

- Tailplane-vertical fin junction.

- Fuselage mesh generation.

- Wing-fuselage junction blocks generation. $x - z$ plane symmetry is exploited.

- Vertical empennage-fuselage connection.

The final number of blocks for the mesh is 23: 3 for each wing, 6 for the fuselage, 2 for each horizontal empennage semi-wing, 2 for the vertical empennage, 2 for each wing-fuselage connection block and 1 for the empennage junction patch. Figure 2.24 shows the V5 "Rondone" full aircraft mesh organised into blocks.

FIGURE 2.24: V5 "Rondone" full aircraft mesh organised into blocks.

# Chapter 3

# COMPA analysis

## 3.1 Mesh sizing and validation

The purpose of this analysis is to establish how much the coefficients computed by COMPA are dependent on the discretization, and decide the number of panels needed for every part.

### 3.1.1 Convergence analysis

Several COMPA runs need to be carried out with different mesh refinements for all of the aircraft parts at different aerodynamic and control angles, representing the different operational conditions. This means that the analysis have to be performed with respect to the variations of 10 independent parameters (5 mesh parameters, 2 aerodynamic angle and 3 control angles), which would lead to an enormous amounts of runs to be carried out with COMPA. This approach is impossible due to the amount of computational time it would take.

In order to simplify the task it is assumed that the various parts can be analysed independently. The analysis is thus only performed on the wing and individual empennages. A Matlab code automates this convergence analysis by launching different COMPA tests on meshes with different mesh parameters and angles. After the wings and the empennage meshes are sized, the fuselage can be tested manually to see the influence of its mesh refinement on the overall coefficients.

The geometry used to test each part includes the part itself and the blocks at its boundaries. The meshes that were tested are shown in Figure 3.1.
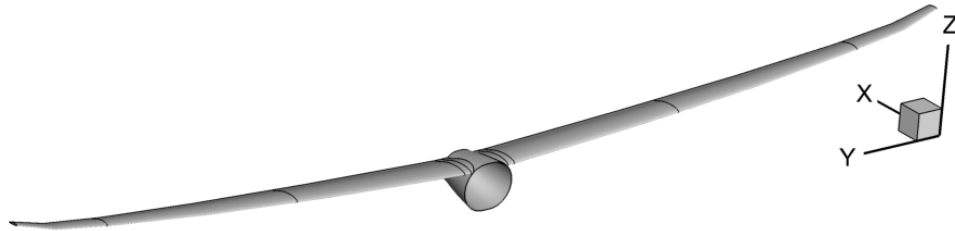


(a) Geometry for the wing.
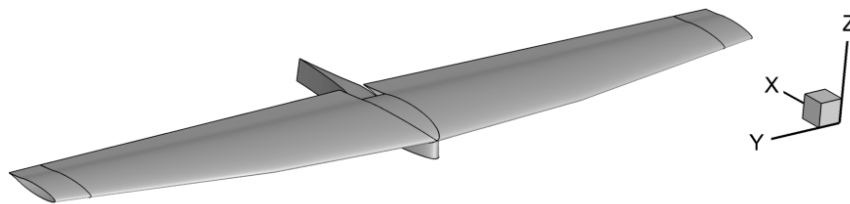


(b) Geometry for the horizontal empennage.
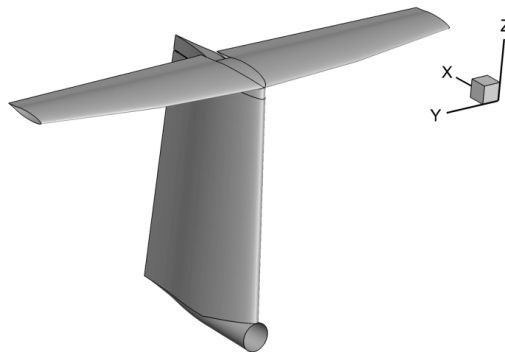


(c) Geometry for the vertical empennage.

FIGURE 3.1: Geometries used for the convergence analyses.

The analysis tested the parts with different combinations of longitudinal and sectional grid refinement: the number of wing panels along both directions was tested with the following values[20,40,60,80,100], for a total of 25 combinations. For the two empennages the test values were limited to [20,40,60,80], for a total of 16 combinations. The same analysis was carried at different values of the AoA $(-5°, 0°, -5°)$, of the AoS $(0°, 5°)$ and of the control surface deflection $(-10°, 0°, 10°)$. The number of tests was 150 for the wing and 96 for each of the two empennages, for a total

of 342. The whole convergence analysis has taken over a week of total computational time.

The results of those calculations constitute a large amount of data. In order to study it, the following variable is used as a measure of the discretization error:

$$\varepsilon_{Cx_i} = \left| \frac{C_{x_i} - C_{x_{MaxPan}}}{C_{x_{MaxPan}}} \right|, \tag{3.1}$$

where $C_{x_i}$ is either the lift or drag coefficient computed with a particular mesh configuration, $\varepsilon_{Cx_i}$ is the error indicator for that configuration and $C_{x_{MaxPan}}$ is the coefficient computed with the most refined mesh of the analysis. This is a relative error, for it to be a usable approximation of the discretization error it must converge in a soft plateau.

The results of the convergence analysis are plotted as a surface in function of the number of section panels on one axis and longitudinal panels on the other.

Figures 3.2 and 3.3 presents the convergence analysis on the wing $C_D$ and $C_L$ in two different configurations among the many tested .

## 3.1.2 Mesh sizing

This tests allowed to choose the grid size for the wing: 60 panels on the airfoil section and 60 on the wing span(30 per semi-wing) would always keep the error on the $C_L$ around 1% (always below 2%) and that on the $C_D$ around 2% (always below 5%). With the same criteria it was chosen the horizontal empennage mesh refinement: 50 panels on the airfoil section and 40 on the span. This implies a number of panels of 70 on the airfoil of the vertical empennage, which was confirmed more than enough by the analysis of that part. Lastly, the number of span panels on the vertical empennage was set at 20.

After the establishment of the mesh refinement for all the lifting surface some manual test on the full geometry varying the fuselage mesh parameters are carried out to establish the lightest refinement for the fuselage that would not affect the outputs. The final mesh is
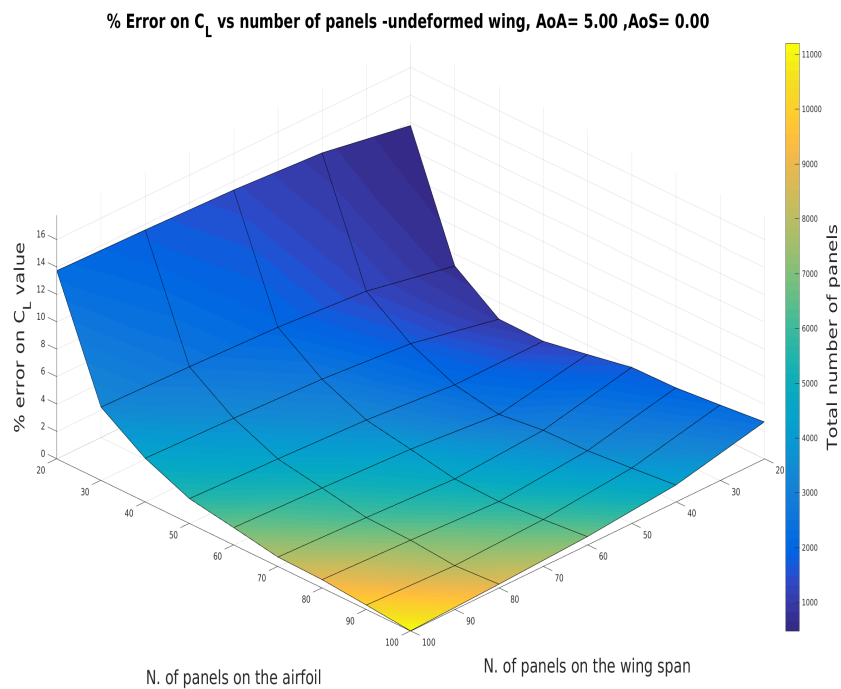
(a) Error on $C_L$.
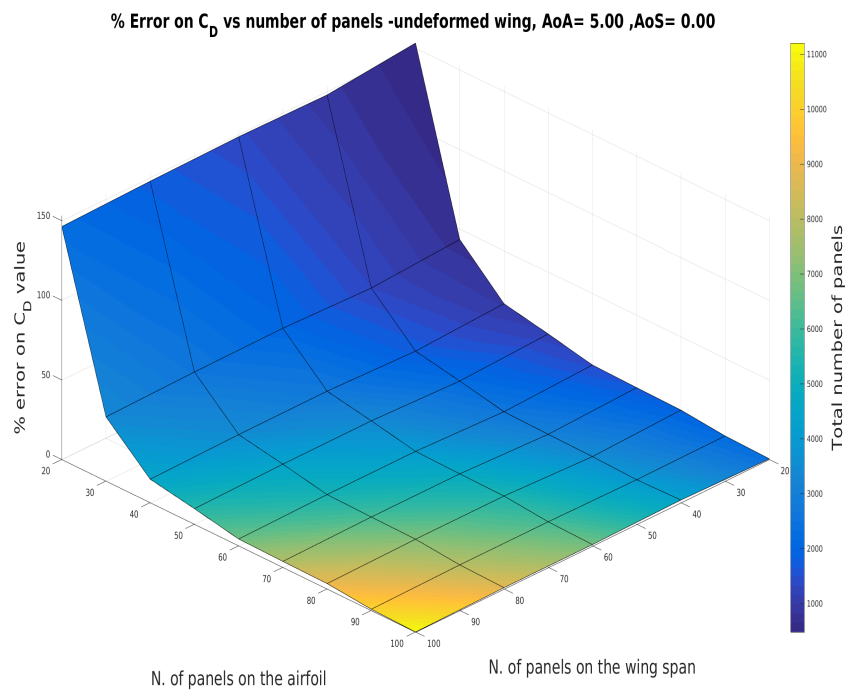


(b) Error on $C_D$.

FIGURE 3.2: Convergence analysis for the wing mesh
at $\alpha = 5°$, $\beta = 0°$ and $\Delta_A = 0°$.

shown in figure 3.4. The total number of mesh panels is 12440, generated by 13862 grid points.
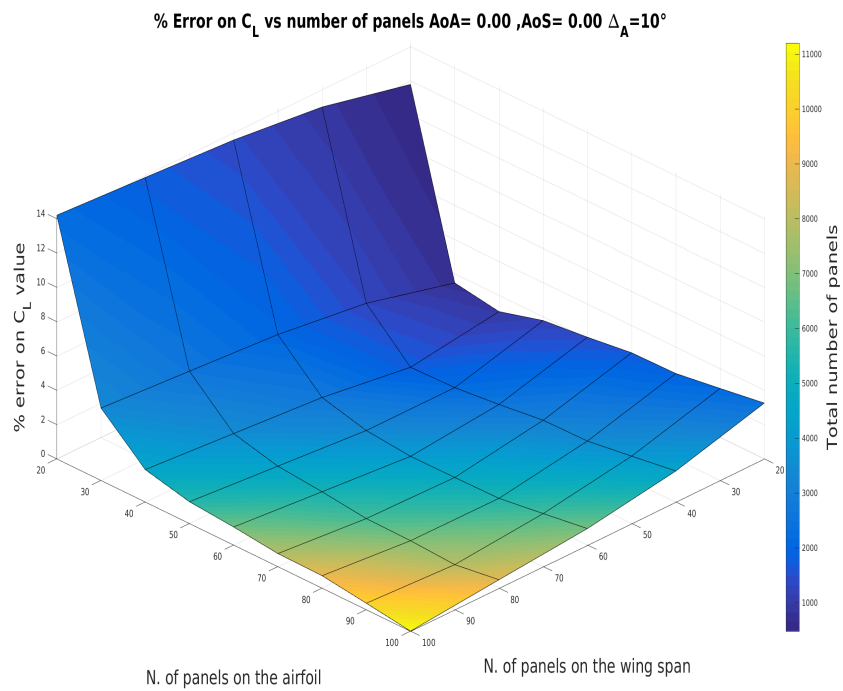
(a) Error on $C_L$.
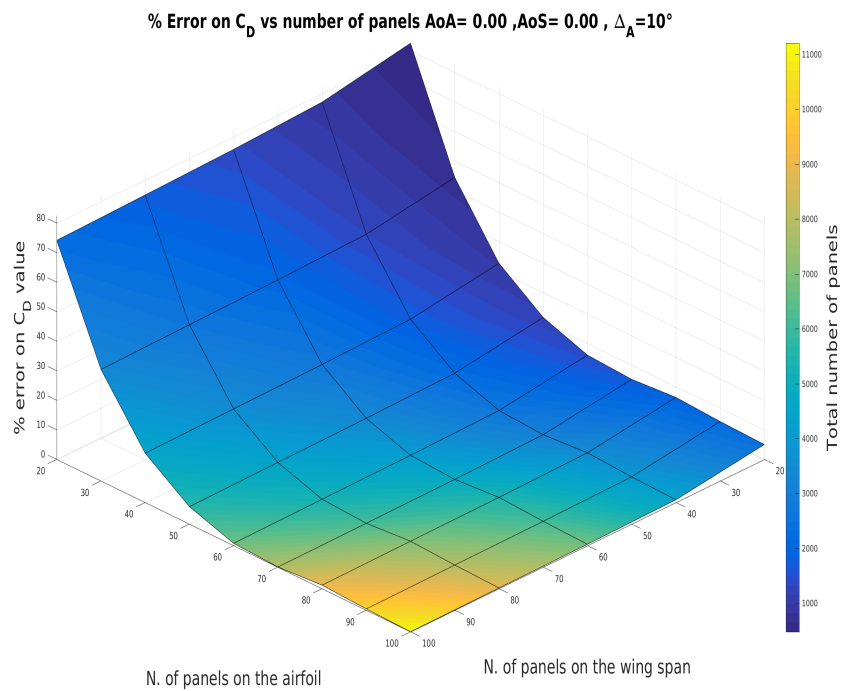


(b) Error on $C_D$.

FIGURE 3.3: Convergence analysis with $\alpha = 0°$ $\beta = 0°$ and $\Delta_A = 10°$.
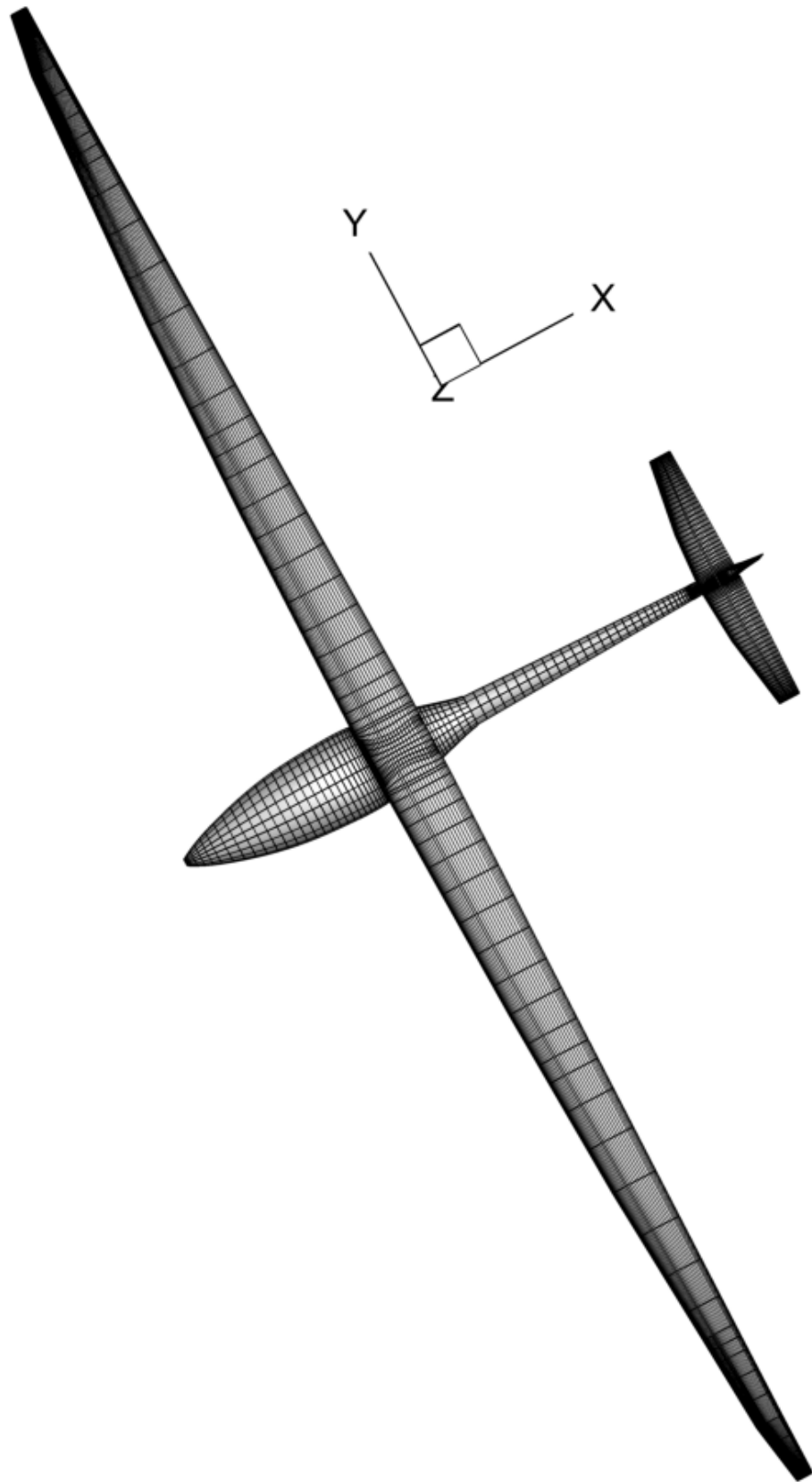
FIGURE 3.4: Final V5 "Rondone" mesh after the con-
vergence analysis.

## 3.2   Full geometry analysis

After establishing the proper number of panels the sailplanes the first batch of full geometry analyses are performed. Those are preliminary analysis needed to asses in which conditions to perform the full geometry simulation. The tests are performed at the following AoAs: $[-5°, -2°, 0°, 2°, 5°]$. The AoS and the ailerons, elevator and rudder deflection angles ($\Delta_A$, $\Delta_E$ and $\Delta_R$) are all set at $0°$ in the first simulation. For the following ones each of them is perturbed in turn by $1°$ and then by $4°$, the elevator is also tested at $-1\circ$

The results of this first batch of analyses are used to compute the untrimmed drag polar (with all control angles set at $0°$), verify the linearies and of the different variables, calculate the aerodynamic center position and its coefficient $C_{My}(A)$ at $\Delta_E = 0$. Lastly, the $C_{L,\alpha}$, $C_{L,\Delta_E}$ and $C_{My,\Delta_E}$ derivatives are computed in order to trim the aircraft and find the trim values of the AoA and $\Delta_E$ for different flight conditions. Those values are then used to make a second batch analysis in order to compute the control and stability derivatives in those flight conditions and the trimmed drag polar.

### 3.2.1   Preliminary analysis

Figures 3.5 and 3.6 show the pressure distributions from two of the test cases of the preliminary analyses.

**Lift and drag curves**

Figure 3.7 shows the lift and drag curves of the V5, generated from the output coefficients of the analysis.

Those curves are consistent with the panel method of COMPA: the $C_L - \alpha$ curve is perfectly linear due to the lack of flow separation. The $C_D - \alpha$ curve is parabolic because the COMPA analysis is inviscid and can only capture the induced drag. The induced drag drag value does not reach 0 as predicted by the flight dynamics theory $\left(C_{Di} = \frac{C_L^2}{\lambda \pi e}\right)$ because it is the sum of the wing and horizontal
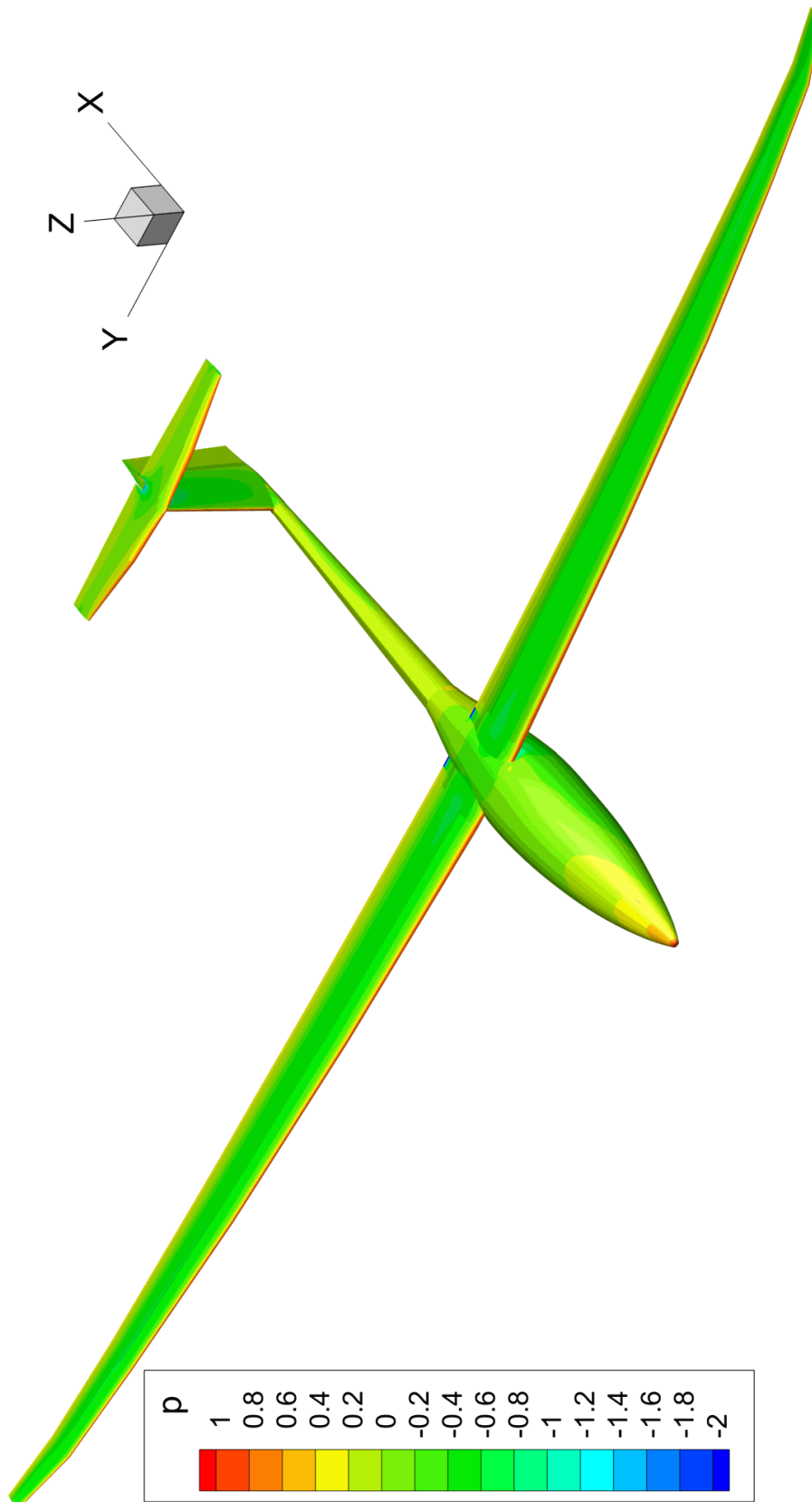
FIGURE 3.5: V5 surface pressure distribution with $\alpha = -5°$, $\beta = 0°$, $\Delta_A = 0°$, $\Delta_E = 0°$, $\Delta_R = 0°$.
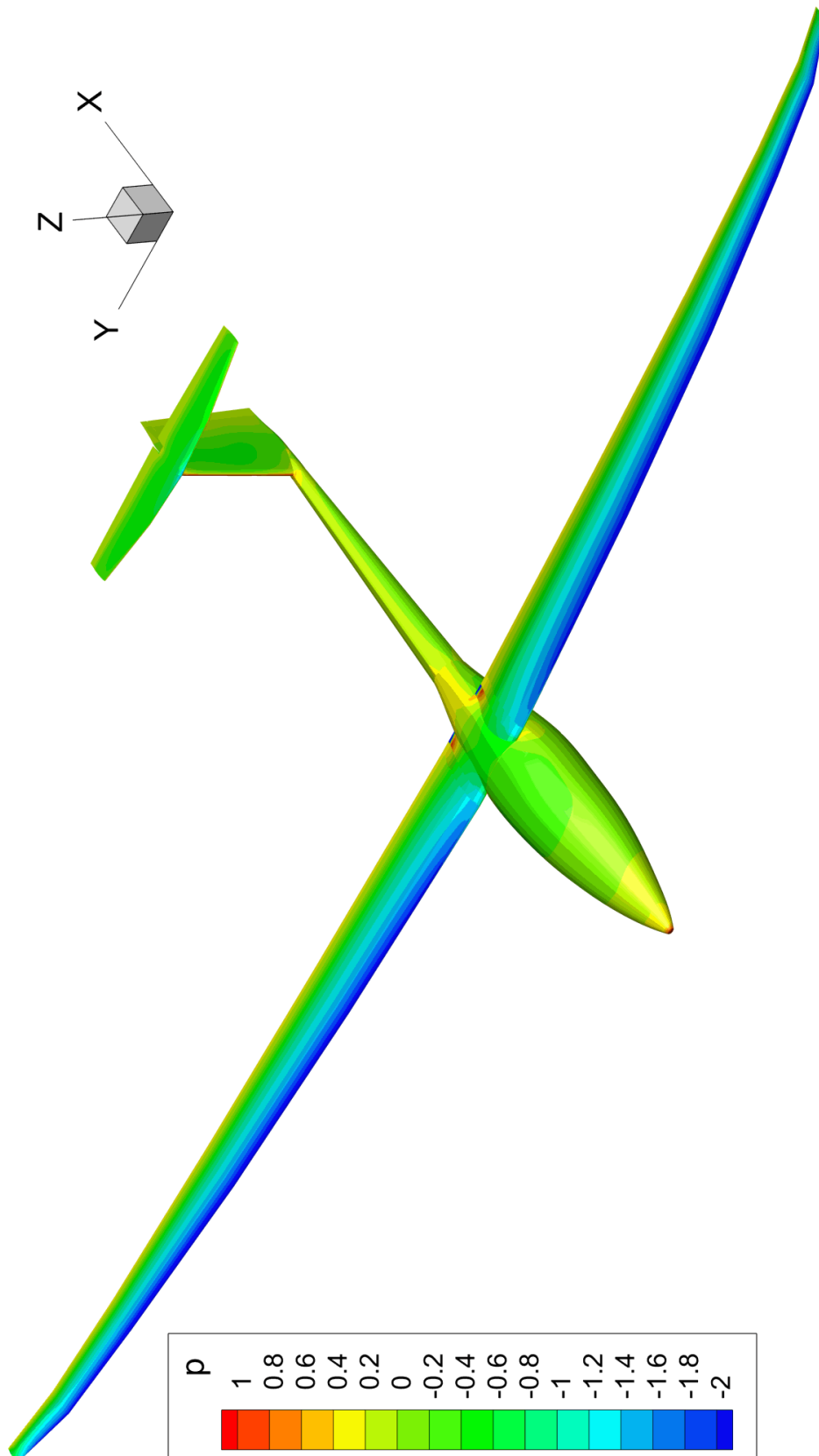
FIGURE 3.6: V5 surface pressure distribution with $\alpha = 5°$, $\beta = 0°$, $\Delta_A = 0°$, $\Delta_E = 0°$, $\Delta_R = 0°$.
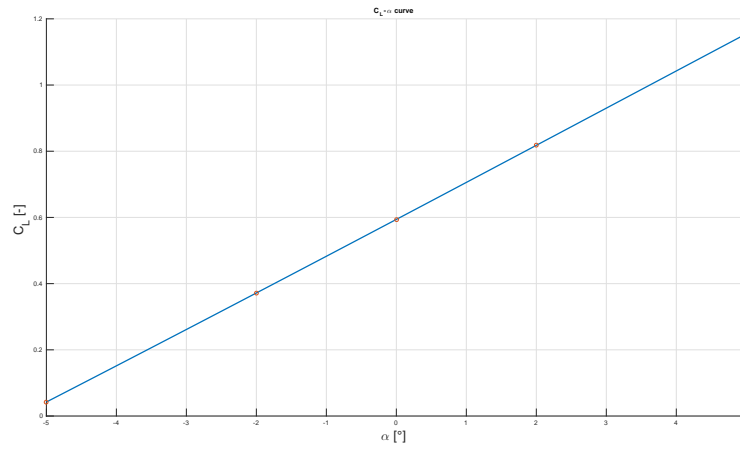
(a) $C_L - \alpha$ curve.
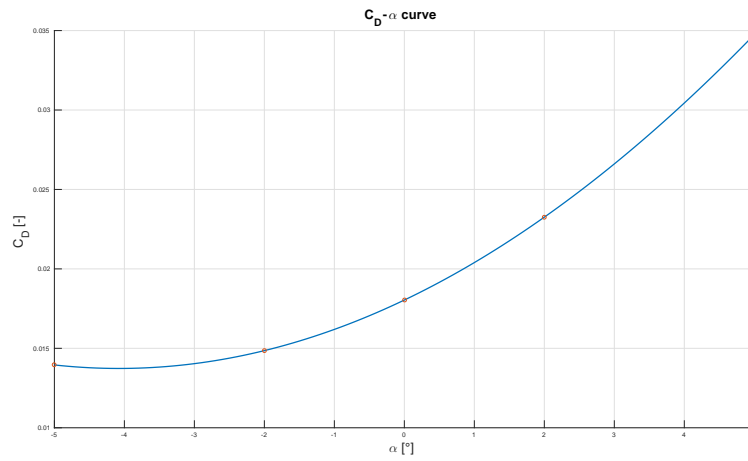


(b) $C_D - \alpha$ curve.
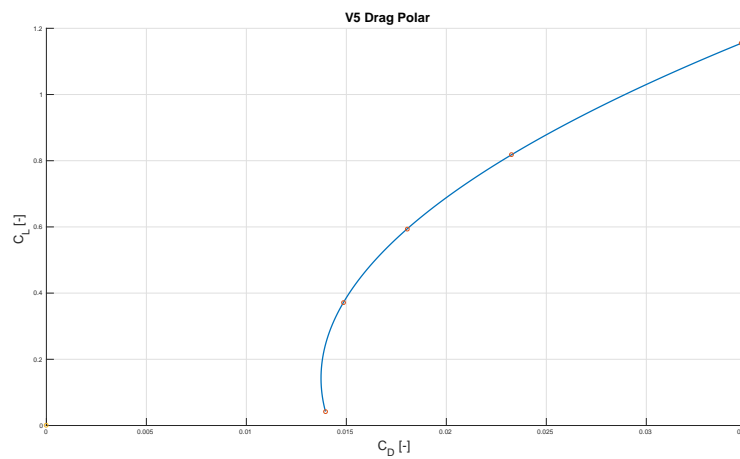


(c) $C_L - C_D$ Drag polar curve.

FIGURE 3.7: V5 "Rondone" lift curve, drag curve and drag polar.

empennage drag coefficients and, being at different pre-pitch angles, they individually reach 0 at different AoA values.

**Aerodynamic center**

According to the traditional flight dynamics theories [21][22] the aerodynamic center is positioned on the longitudinal axis of the aircraft and is computed as:

$$x_A = -C_{My,\alpha}/C_{z,\alpha}, \tag{3.2}$$

here $C_z$ is the vertical force coefficient in the body reference frame. $C_{My,\alpha}$ and $C_{z,\alpha}$ are obtained with a linear regression of the computed $C_{My}$ and $C_z$ values along the AoA values for the COMPA tests.

When the moment coefficient is calculated with respect to this point the resulting $C_{My}(x_A) - \alpha$ curve, shown in figure 3.8, is noticeably non-linear.
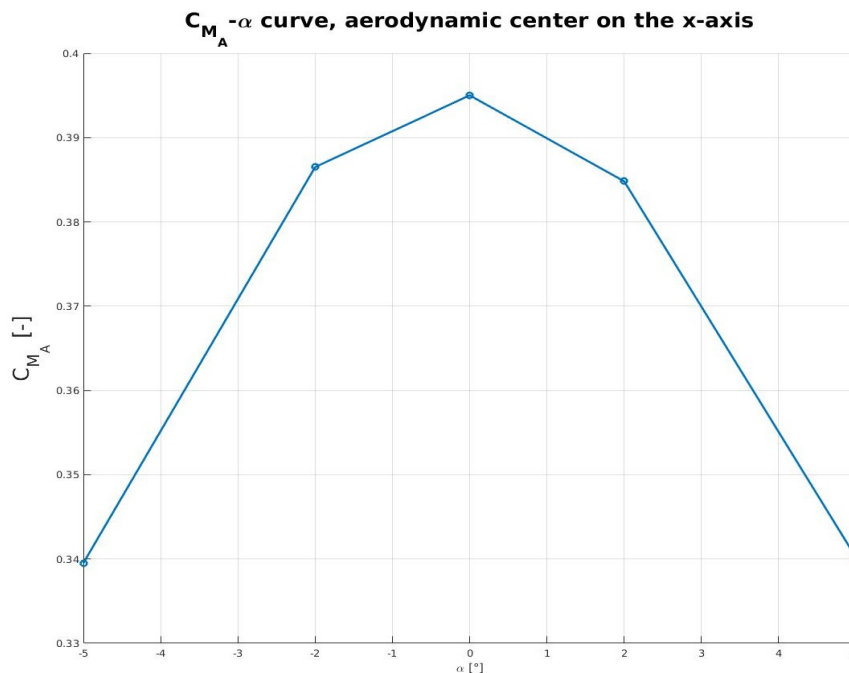


FIGURE 3.8: V5 "Rondone" $C_{My} - \alpha$ curve, with $C_{My}$ computed with respect to the aerodynamic center on the x-axis.

The non-linearity of the $C_{My} - \alpha$ curve is a direct consequence of the non-linearity of the $C_D - \alpha$ curve. Equation 3.2 is inaccurate because it is based on the assumption that the drag is either negligible, linear

in $\alpha$ or aligned with the aerodynamic center (thus not producing any moment). A more modern concept of aerodynamic center that takes into account the non-linearities is proposed by Phillips, Alley and Niewoehner [23]. In their model the aerodynamic center position on the aircraft longitudinal axis dependent on the AoA (in fact, the point computed with Equation 3.2 is valid at $\alpha = 0°$).

By expanding the concept of the aerodynamic center to the $x - z$ plane it is possible to find a point where the moment can be considered independent from the AoA with a good degree of accuracy[23].

In order to find this point an iterative method is implemented:

0. Guess a point $A : [x_A^{(0)}, z_A^{(0)}]$. $x_A^{(0)}$ is chosen with the 3.2 while $z_A^{(0)} = 0$ .

1. Moment coefficient at computation for all the AoA in the analysis $\{C_{My}(A)\}^{(k)} = \{C_{My}(0)\} + x_A^{(k)}\{C_z\} - z_A^{(k)}\{C_x\}$

2. Quadratic regression of $\{C_{My}(A)\}^{(k)}$ in $\alpha$, the coefficient of the quadratic term is used as the error $\varepsilon^{(k)}$ on $z_A^{(k)}$

3. Adjustment of $z_A$: $z_A^{(k+1)} = z_A^{(k)} - k\varepsilon^{(k)}$

4. Update of the moment coefficient:
   $\{\overline{C}_{My}(A)\}^{(k+1)} = \{C_{My}(0)\} + x_A^{(k)}\{C_z\} - z_A^{(k+1)}\{C_x\}$

5. Linear regression of $\{\overline{C}_{My}(A)\}^{(k+1)}$ in $\alpha$ to estimate $\overline{C}_{My,\alpha}^{(k+1)}$

6. Correction of $x_A$: $x_A^{(k+1)} = x_A^{(k)} - \overline{C}_{My,\alpha}^{(k+1)}/C_{z,\alpha}$

7. If $|\varepsilon^{(k)}| < \varepsilon_{min}$ then end the cycle, else go back to point 1.

The outcome of this iterative process is shown in Figure 3.9: as it can be seen the new aerodynamic centre point on the $x - z$ plane greatly reduces non linearities

**Elevator deflection**

In order to use the traditional flight dynamics formulae to trim the sailplane it needs to be verified that the deflection of the elevator does not have any effect on the aerodynamic centre. Figure 3.10 shows how the $C_{My,\alpha}(A)$ curves changes with different values of
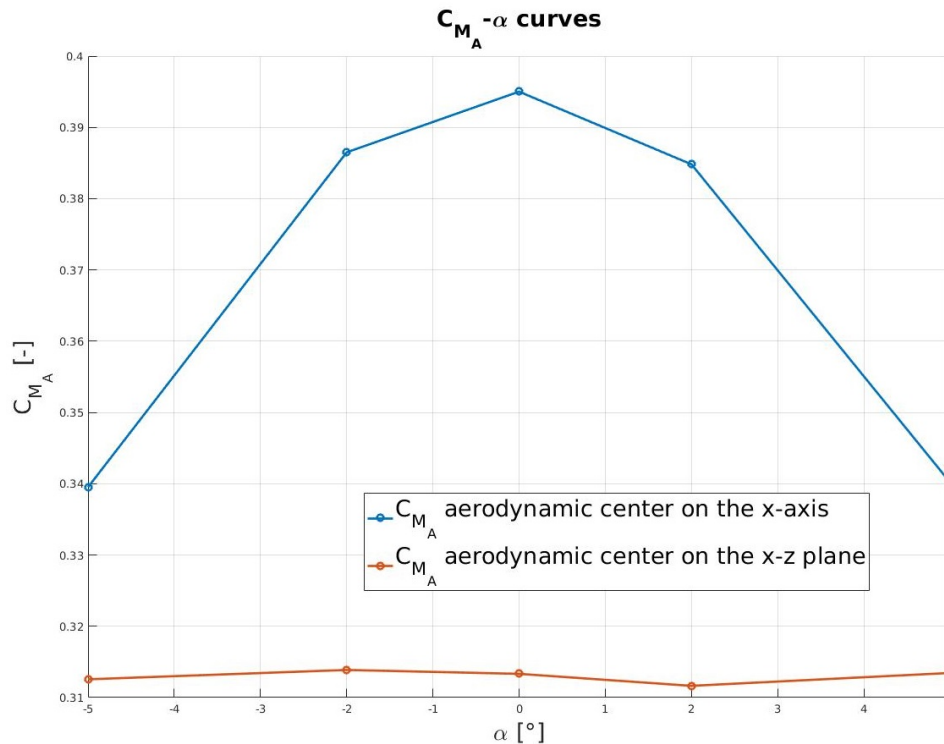
FIGURE 3.9: Comparison between $C_{My} - \alpha$ curves linearity depending on the aerodynamic center position constraints.

$\Delta_E$. It can be be safely assumed that the aileron deflection does not affect the slope on the curves in a relevant way, so the aerodynamic center is fixed for every value of interest of $\Delta_E$ and $\alpha$

The last important thing to check before trimming the aircraft is the linearity of the $C_{My}(A) - \Delta_E$ and $C_L - \Delta_E$ curves, in order to compute the derivatives $C_{My,\Delta_E}$ and $C_{L,\Delta_E}$. Figures 3.11(a) and 3.11(b) show that both of them are perfectly linear.

**Trim**

Trimming an aircraft means finding the $\alpha$ and $\Delta_E$ values that enable steady level flight in a particular flight condition. Every flight condition is defined by the sailplane equivalent air speed, $v_e =$EAS, weight $W$, and longitudinal center of gravity coordinate $x_g$. Each flight condition is equivalent to a trim lift and moment coefficient from vertical forces equilibrium and moments equilibrium on the
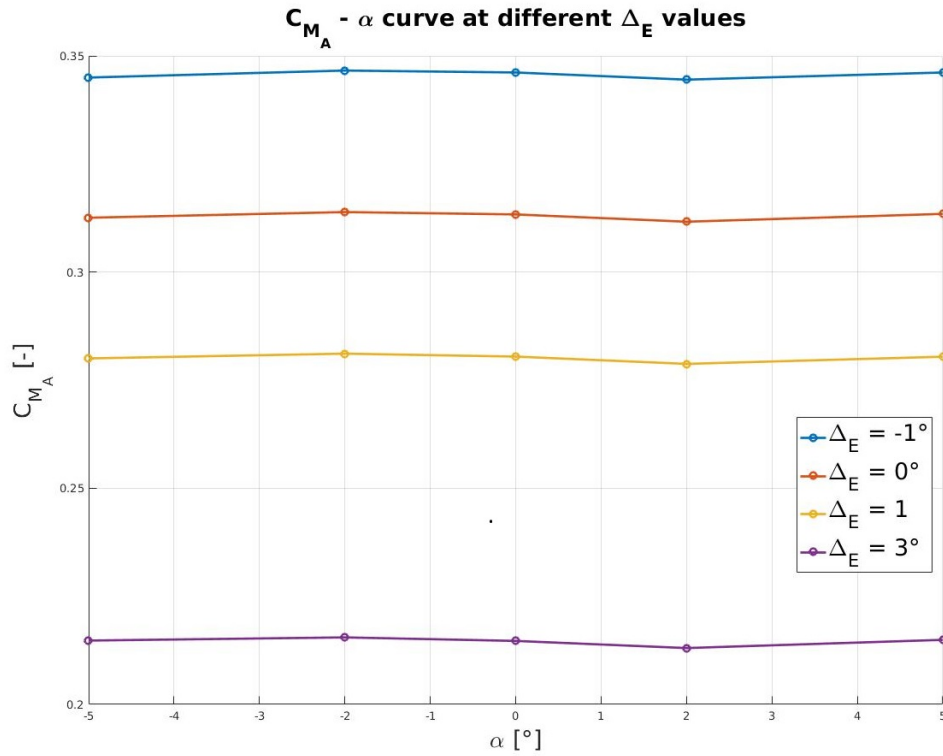
FIGURE 3.10: Effects of the elevator deflection on the slope of $C_{My} - \alpha$ curves.

pitch axis:

$$C_L|_{trim} = \frac{2W}{v_e^2 S}, \tag{3.3}$$

$$C_{My}(A)|_{trim} = \frac{2Wd(\alpha)}{v_e^2 cS}, \tag{3.4}$$

where S is the planar surface of the wing, $c$ is the mean aerodynamic chord and $d(\alpha)$ is the lever arm between the weight and the aerodynamic center when the centre of gravity and the aerodynamic centre are not both on the longitudinal axis. $d(\alpha)$ is represented in Figure 3.12.

The expression for $d(\alpha)$ is:

$$d(\alpha) = (x_G - x_A)\cos(\alpha) - (z_G - z_A)\sin(\alpha) \tag{3.5}$$

This term could be approximated for small $\alpha$ as:
$d(\alpha) \approx (x_G - x_A) - (z_G - z_A)\alpha$ , but in order to simplify the trim calculations it would be ideal to eliminate the dependency of $C_{My}(A)|_{trim}$ from $\alpha$. In order to do this the aircraft is trimmed in two steps where in the first step $d(\alpha)$ is approximated as $\tilde{d} = x_G - x_A$
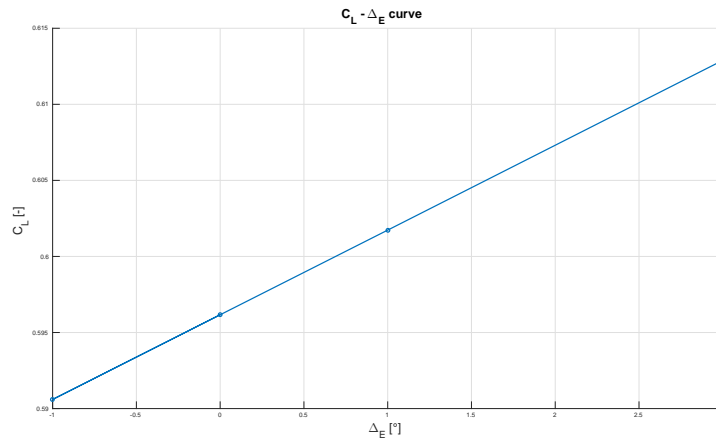
(a) $C_L - \Delta_E$ curve.
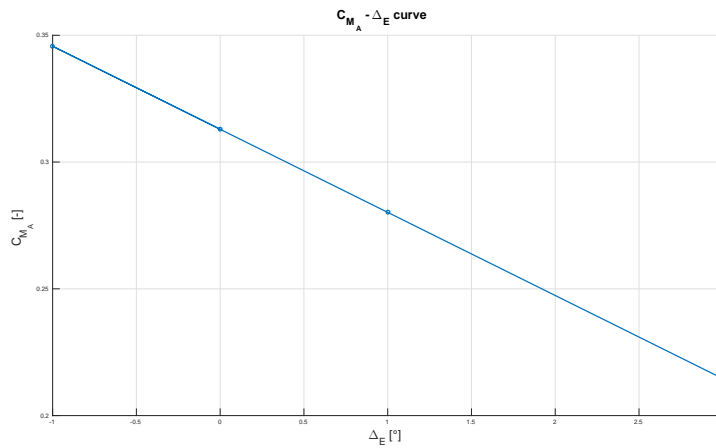


(b) $C_{My} - \Delta_E$ curve.

FIGURE 3.11: Linearity of the elevator deflection effects on $C_L$ and $C_{My}$.
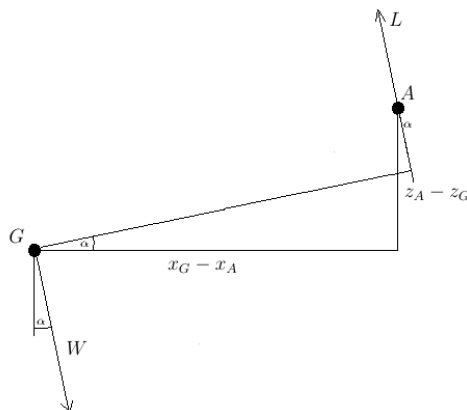


FIGURE 3.12: $d(\alpha)$ geometric representation.

(distance between the projections of the aerodynamic centre and centre of gravity on the longitudinal axis), while the second step is

used to refine the results with the exact formulations:

$$\tilde{C}_{My}(A)|_{trim} = \frac{2W\tilde{d}}{v_e^2 cS}, \tag{3.6}$$

$$\tilde{\Delta}_E|_{trim} = \frac{\tilde{C}_{My}(A)|_{trim} - C_{My}(A)|_{\Delta_E=0}}{C_{My,\Delta_E}}, \tag{3.7}$$

$$\tilde{\alpha}|_{trim} = \frac{C_L|_{trim} - C_{L,\Delta_E}\tilde{\Delta}_E - C_L|_{\Delta_E=0}^{\alpha=0}}{C_{L,\alpha}}, \tag{3.8}$$

$$C_{My}(A)|_{trim} = \frac{2Wd(\tilde{\alpha}|_{trim})}{(v_e^2 cS}, \tag{3.9}$$

$$\Delta_E|_{trim} = \frac{C_{My}(A)|_{trim} - C_{My}(A)|_{Delta_E=0}}{C_{My,\Delta_E}}, \tag{3.10}$$

$$\alpha|_{trim} = \frac{C_L|_{trim} - C_{L,\Delta_E}\Delta_E - C_L|_{\Delta_E=0}^{\alpha=0}}{C_{L,\alpha}}. \tag{3.11}$$

The second COMPA analysis, which is performed on the trim configuration, is used to confirm the accuracy of the trim angles computed with this procedure.

During and after the computation of the trim angles for a given condition there are some conditions that needs to be verified to make sure that flight is possible in that configuration. Specifically, flight is impossible if:

- $C_L|_{trim} > C_{Lmax}$: stall.

- $|\Delta_E|_{trim}| > \Delta_{Emax}$: center of gravity position too forward.

The conditions at which the sailplane is trimmed are chosen at the 8 edges of the space defined by the permissible values of the $EAS$, weight and center of gravity coordinate

### 3.2.2 Second analysis

The second analysis is performed by running, for every flight condition from Tables 3.1 and 3.2 , a test in the condition itself and a

TABLE 3.1: Trimmed flight conditions for the second analysis, conditions 1-5.

| Condition | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| EAS [km/h] | 110 | 120 | 270 | 90 | 270 |
| W[kg] | 440 | 740 | 740 | 430 | 430 |
| $x_g$[m] | 2.71 | 2.51 | 2.51 | 2.51 | 2.51 |
| $\alpha\vert_{trim}[°]$ | 0.32 | 3.24 | -4.05 | 3.54 | -4.81 |
| $\Delta_E\vert_{trim}[°]$ | 0 | -11.91 | 5.31 | -12.62 | 7.08 |

TABLE 3.2: Trimmed flight conditions for the second analysis, conditions 6-9.

| Condition | 6 | 7 | 8 | 9 |
|---|---|---|---|---|
| EAS [km/h] | 110 | 270 | 80 | 270 |
| W[kg] | 740 | 740 | 430 | 430 |
| $x_g$[m] | 2.91 | 2.91 | 2.91 | 2.91 |
| $\alpha\vert_{trim}[°]$ | 4.01 | -4.21 | 4.97 | -4.91 |
| $\Delta_E\vert_{trim}[°]$ | 2.95 | 8.45 | 2.30 | 8.90 |

test for each relevant angle ($\alpha$,$\beta$,$\Delta_E$,$\Delta_A$ and $\Delta_R$) perturbed by $1°$. The control and stability derivatives are computed with a finite difference derivative between the perturbed and the original value of the parameter. In this analysis the control angles are set as the actual deflection angle of the surfaces they refer to, except for the ailerons. The ailerons work asymmetrically: the aileron that moves up (decreasing lift) is deflected by an angle that is double of the one that moves down (increasing lift). The value of the angle is then set equal to the smallest of the to deflections (as absolute value) and is set to be positive for a right-turning deflection.

TABLE 3.3: Longitudinal derivatives, conditions 1-5.

| Condition | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $C_{L,\alpha}$ | 6.47 | 6.48 | 6.40 | 6.48 | 6.39 |
| $C_{D,\alpha}$ | 0.087 | 0.133 | -0.002 | 0.138 | -0.016 |
| $C_{My,\alpha}$ | -3.583 | -6.136 | -4.564 | -6.195 | -4.438 |
| $C_{L,\Delta_E}$ | 0.297 | 0.288 | 0.297 | 0.287 | 0.297 |
| $C_{D,\Delta_E}$ | -0.005 | -0.014 | -0.008 | -0.014 | -0.008 |
| $C_{My,\Delta_E}$ | -2.021 | -2.089 | -2.073 | -2.087 | -2.062 |

Tables 3.3 and 3.3 report the longitudinal derivatives. The main thing to note here is the non-linearity of $C_{My,\alpha}$, which in configuration nine, with an aft center of gravity and a high speed,

TABLE 3.4: Longitudinal derivatives, conditions 6-9.

| Condition | 6 | 7 | 8 | 9 |
|---|---|---|---|---|
| $C_{L,\alpha}$ | 6.49 | 6.40 | 6.49 | 6.39 |
| $C_{D,\alpha}$ | 0.177 | 0.001 | 0.197 | -0.014 |
| $C_{My,\alpha}$ | -2.401 | -0.589 | -2.599 | -0.479 |
| $C_{L,\Delta_E}$ | 0.292 | 0.297 | 0.292 | 0.297 |
| $C_{D,\Delta_E}$ | 0.006 | -0.004 | 0.007 | -0.005 |
| $C_{My,\Delta_E}$ | -1.954 | -1.878 | -1.928 | -1.870 |

TABLE 3.5: Lateral derivatives, conditions 1-5.

| Condition | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $C_{Y,\beta}$ | 0.169 | 0.118 | 0.317 | 0.116 | 0.334 |
| $C_{Mx,\beta}$ | -0.219 | -0.203 | -0.241 | -0.202 | -0.244 |
| $C_{Mz,\beta}$ | 0.113 | 0.050 | 0.272 | 0.047 | 0.292 |
| $C_{Y,\Delta_A}$ | 0.104 | 0.101 | 0.108 | 0.100 | 0.109 |
| $C_{Mx,\Delta_A}$ | -0.945 | -0.943 | -0.942 | -0.942 | -0.941 |
| $C_{Mz,\Delta_A}$ | 0.064 | 0.020 | 0.126 | 0.016 | 0.137 |
| $C_{Y,\Delta_R}$ | 0.217 | 0.218 | 0.214 | 0.218 | 0.212 |
| $C_{Mx,\Delta_R}$ | -0.017 | -0.017 | -0.018 | -0.017 | -0.018 |
| $C_{Mz,\Delta_R}$ | 0.218 | 0.215 | 0.210 | 0.215 | 0.208 |

TABLE 3.6: Lateral derivatives, conditions 6-9.

| Condition | 6 | 7 | 8 | 9 |
|---|---|---|---|---|
| $C_{Y,\beta}$ | 0.117 | 0.320 | 0.117 | 0.335 |
| $C_{Mx,\beta}$ | -0.199 | -0.241 | -0.194 | -0.244 |
| $C_{Mz,\beta}$ | 0.052 | 0.287 | 0.049 | 0.306 |
| $C_{Y,\Delta_A}$ | 0.100 | 0.109 | 0.099 | 0.109 |
| $C_{Mx,\Delta_A}$ | -0.941 | -0.942 | -0.939 | -0.941 |
| $C_{Mz,\Delta_A}$ | 0.013 | 0.133 | -0.001 | 0.143 |
| $C_{Y,\Delta_R}$ | 0.206 | 0.211 | 0.203 | 0.210 |
| $C_{Mx,\Delta_R}$ | -0.015 | -0.018 | -0.015 | -0.018 |
| $C_{Mz,\Delta_R}$ 8 | 0.211 | 0.216 | 0.208 | 0.215 |

almost leads to the loss of static stability.$C_{My,\Delta_E}$ is the elevator sensitivity.

Tables 3.5 and 3.6 report the lateral derivatives. $C_{Mx,beta}$ is the dihedral effect, where a positive $\beta$ induces a negative $C_{Mx}$,$C_{Mx,\Delta_A}$ is the ailerons sensitivity, which is a little low, but it comes as no surprise given the small size of the ailerons on the V5. $C_{Mz,\Delta_A}$ is positive and represents the tendency of the plane to yaw on the opposite side of the direction the pilot is trying to roll towards. It is that this term is small. Finally, $C_{Mz,\Delta_R}$, the rudder sensitivity, is

TABLE 3.7: Mixed longitudinal-lateral derivatives, conditions 1-5.

| Condition | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $C_{L,\beta}$ | -0.015 | -0.022 | -0.007 | -0.022 | -0.006 |
| $C_{D,\beta}$ | 0.002 | 0.001 | 0.002 | 0.001 | 0.002 |
| $C_{My,\beta}$ | 0.025 | 0.043 | 0.017 | 0.048 | 0.019 |
| $C_{L,\Delta_A}$ | -0.560 | -0.560 | -0.557 | -0.560 | -0.557 |
| $C_{D,\Delta_A}$ | -0.004 | -0.007 | 0.001 | -0.007 | 0.001 |
| $C_{My,\Delta_A}$ | 0.201 | 0.401 | 0.332 | 0.409 | 0.327 |
| $C_{L,\Delta_R}$ | -0.001 | -0.001 | 0.000 | -0.001 | -0.001 |
| $C_{D,\Delta_R}$ | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| $C_{My,\Delta_R}$ | 0.005 | 0.003 | 0.005 | 0.011 | 0.005 |

TABLE 3.8: Mixed longitudinal-lateral derivatives, conditions 5-9.

| Condition | 6 | 7 | 8 | 9 |
|---|---|---|---|---|
| $C_{L,\beta}$ | -0.029 | -0.007 | -0.029 | -0.006 |
| $C_{D,\beta}$ | 0.001 | 0.002 | 0.000 | 0.002 |
| $C_{My,\beta}$ | 0.051 | 0.014 | 0.062 | 0.017 |
| $C_{L,\Delta_A}$ | -0.562 | -0.557 | -0.562 | -0.557 |
| $C_{D,\Delta_A}$ | -0.007 | 0.001 | -0.008 | 0.001 |
| $C_{My,\Delta_A}$ | 0.051 | -0.013 | 0.088 | -0.018 |
| $C_{L,\Delta_R}$ | -0.001 | -0.001 | 0.000 | -0.001 |
| $C_{D,\Delta_R}$ | 0.001 | 0.001 | 0.001 | 0.001 |
| $C_{My,\Delta_R}$ | 0.005 | 0.003 | 0.005 | 0.011 |

small but it is compensated by the fact that the rudder has a high maximum deflection (30°).

Tables 3.7 and 3.8 report the cross-influence terms between the lateral perturbations and the longitudinal coefficients, they are all quite low except for the force and moment induced by the ailerons which, due to their asymmetric deflection, have a relevant impact on the $C_L$ and $C_{Mx}$

# Chapter 4

# Viscous drag estimation

The estimation of the viscous drag coefficient is needed to deepen the knowledge of the V5 flight characteristics. There are different ways to make an estimate of the viscous drag $C_{Dv}$ , the purpose of this chapter is to compare and evaluate them.

## 4.1   Preliminary design formula

This approach is proposed by J. Roskam[1][18] and D.P. Raymer [19] for the estimation of an the zero-lift drag $C_{D0}$ when the project is still in the early stages and many parameters are sill missing. The formula is:

$$C_{D0} = C_{fe}\frac{S_{wet}}{S},\tag{4.1}$$

where $S_{wet}$ is the total wet surface, $C_{fe}$ is the "equivalent skin friction coefficient" and $S$ is the reference area for all coefficients.

$S_{wet}$ is computed very accurately by summing the areas of every panel in the existing mesh for COMPA: for a panel of edges $ABCD$, the area is computed as the sum of the triangles $DAB$ and $BCD$. The area of the triangles is computed as half of the vector product of two of its sides:

$$S_{ABCD} = \frac{1}{2}\left(\parallel \overrightarrow{AB} \times \overrightarrow{AD} \parallel + \parallel \overrightarrow{CB} \times \overrightarrow{CD} \parallel\right).\tag{4.2}$$

The equivalent skin friction coefficient $C_{fe}$ is a parameter dependent on the aircraft type and is based on empirical data. It takes into account all the types of resistances that make the $C_{D0}$ term, on a sailplane those are limited to: skin friction drag, pressure drag and

interference drag. The assumption behind it is that the aircraft geometry is well designed, enough to minimize non-skin friction drag components.

The main issue with using this approach is the little information and the great uncertainty around the value of $C_{fe}$ for sailplanes in literature. Roskam puts it at 0.003 [18] and that is the value that is used here for this method.

## 4.2 Zero-lift drag build-up from individual components - empirical approach

This is the most popular approach to non-computer based estimation of the $C_{D0}$ in the traditional perforance analysis litterature (see Ref, 1, 18, 20 and19). The formulation reported here is that of D.P. Raymer [19] and M. Sadrey [20]. It Computes the overall $C_{D0}$ by computing the skin friction drag for every component and correcting it with additional parameters to account for other kinds of parasitic drags:

$$C_{D0_{subsonic}} = \Sigma_i^n \frac{C_{f_i} FF_i Q_i S_{wet_i}}{S}, \tag{4.3}$$

where $C_{f_i}$ is the flat plate skin friction coefficient for component $i$ and is equal to:

$$C_{f_L} = \frac{1.327}{\sqrt{Re}}, \qquad \text{for laminar boundary layer}, \tag{4.4a}$$

$$C_{f_T} = \frac{0.455}{log_{10}(Re)^{2.58}}, \qquad \text{for turbulent boundary layer}. \tag{4.4b}$$

The final value is a combination between the two:

$$C_{f_i} = kC_{f_L} + (1-k)C_{f_T}. \tag{4.5}$$

$FF_i$ is the form factor and for lifting surfaces at small Mach numbers is equal to:

$$FF_i = 1 + \frac{0.6}{(x/c)_m} \frac{t}{c} + 100 \left(\frac{t}{c}\right)^4.$$ (4.6)

$(x/c)_m$ is the normalized coordinate of the airfoil maximum thickness. For the empennages this form factor needs to be augmented by 10% to take into account the effects of the gap with the hinged control surfaces.

For the fuselage $FF_i$ becomes:

$$FF_i = 1 + \frac{60}{f^3} + \frac{f}{400},$$ (4.7)

Where $f = \frac{L}{\sqrt{(4/\pi)A_{max}}}$.

$Q_i$ is the interference factor for the component $i$ and is negligible for a well designed wing-fuselage intersection, while is about 4% to 5% for a well designed conventional tail configuration.

The problem with this approach is estimating the percentage of laminar flow on the aircraft components. For most aircrafts the flow is mostly turbulent but for a clean geometry such as that of a modern sailplane the laminar flow can reach up to 50% of the surface and it is impossible to asses without using experimental techniques or software currently not available for this project.

## 4.3   Interpolation of 2D airfoil drag polar curves

An alternative approach is to use a quasi 3D method to integrate the viscosity effects in a potential-flow code. In this section Xflr5 iterative LLT/Xfoil method is used for the $C_{Dv}$ estimation.

This operation is performed for the wing, the horizontal empennage and the vertical empennage independently, the viscous drag components are then summed, with the inclusion of the empennage interference factor. This method lacks a fuselage drag estimation. In order to include it, Equation (4.3) was used. For that, a rough
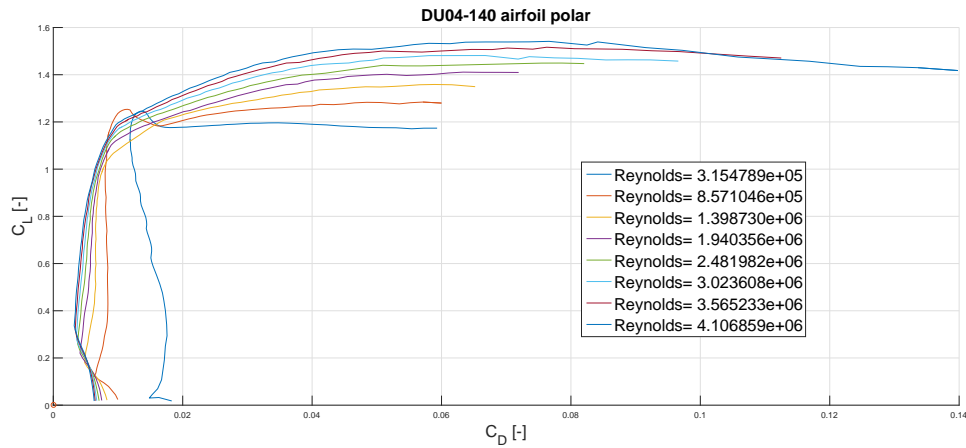
FIGURE 4.1: Family of Xfoil wing polars used for the
Xflr5 LLT analysis.

estimate of the boundary layer configuration on the fuselage is
needed. This was taken from a CFD study by T Hansen [24] on a
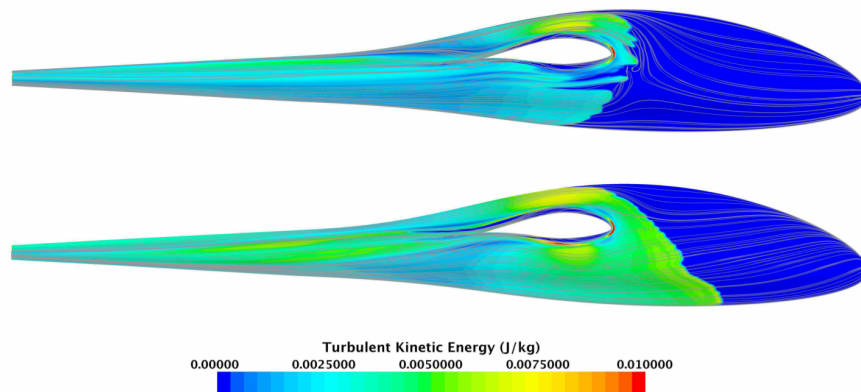sailplane similar in size and geometry to the V5 "Rondone".



FIGURE 4.2: Boundary layer configuration from the
sailplane Standard Cirrus [24].

As it can be seen in Figure 4.2, the transition between laminar and
turbulent boundary layer on the fuselage, although dependent on
the flight speed, seems to happen around the wing root leading
edge coordinate. For this reason the same will be assumed for the
fuselage of the V5.

Figure 4.3 shows a comparison between the different methods to
compute the viscous drag components. The two cases with 100%
and 0% laminar boundary are limit cases and they appear in the
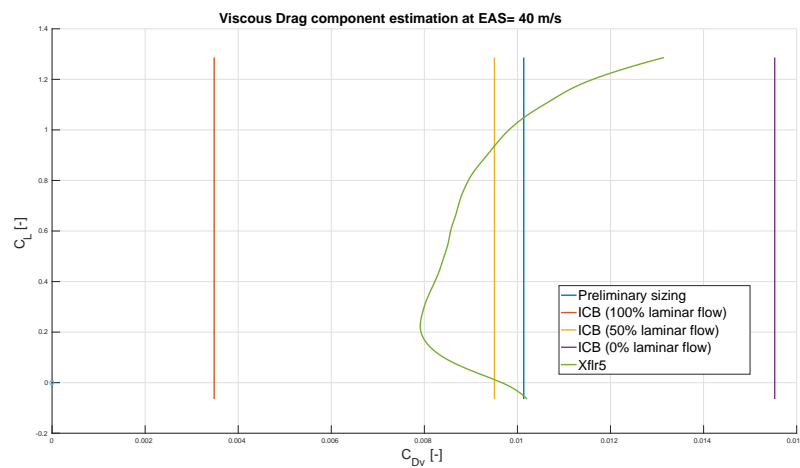plot to provide a visual on how much the results from the

FIGURE 4.3: Viscous drag component computation methods comparison: Preliminary sizing formula, Individual component build-up (ICB) and Xflr5 (with ICB for fuselage).

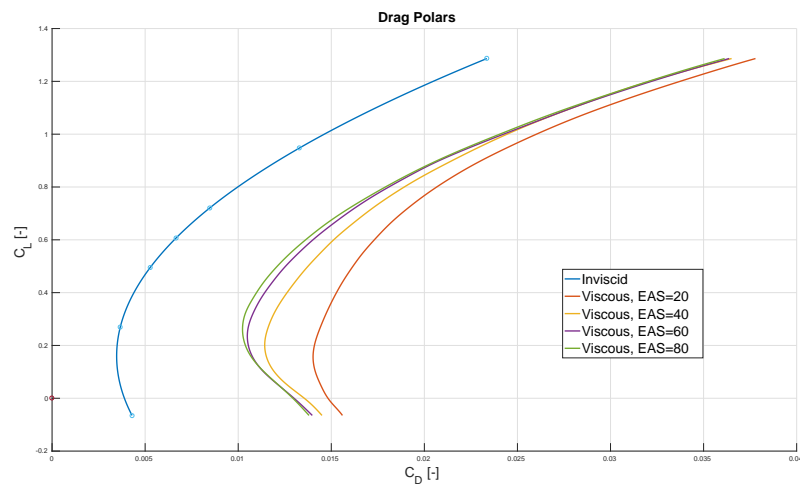individual component build-up may vary depending on the assumed boundary layer type.



FIGURE 4.4: Viscous drag polar correction.

Figure 4.4 shows the viscous drag polar correction at different EAS values using the Xflr5-LLT method.

## 4.4 Viscous trimmed polar

The trimmed polar curve is computed for 440kg of flight weight and center of gravity at 2.71 m from the nose.

The computation starts from the untrimmed inviscid drag polar from the COMPA analysis. In this problem $\alpha|_{trim}$ is already known the unknows are $\Delta_E|_{trim}$ and $v_E|_{trim}$. A first estimate of $v_E|_{trim}$ at each point is computed using the vertical axis equilibrium, $C_{My}|_{trim}$ and consequently $\Delta_E|_{trim}$ are computed using the equilibrium on the $y$ axis moments. Thanks to $\Delta_E|_{trim}$, $C_L|_{trim}$ is computed and may be used for a second iteration to refine the results with a better estimate of $v_E|_{trim}$

The values of $C_{Di}|_{trim}$ are computed with a COMPA analysis at $\alpha|_{trim}$ and $\Delta_E|_{trim}$. At last, the viscous correction is computed with a 2D interpolation of the available viscous polar curves at the required $C_L$ and speed values. For completeness, an estimation of the stall point using Roskam [1] was also carried:

$$C_{L_{max}} = \frac{(C_{L2D_{max}}|_{wingroot} + C_{L2D_{max}}|_{wingtip})/2}{1.05}. \qquad (4.8)$$
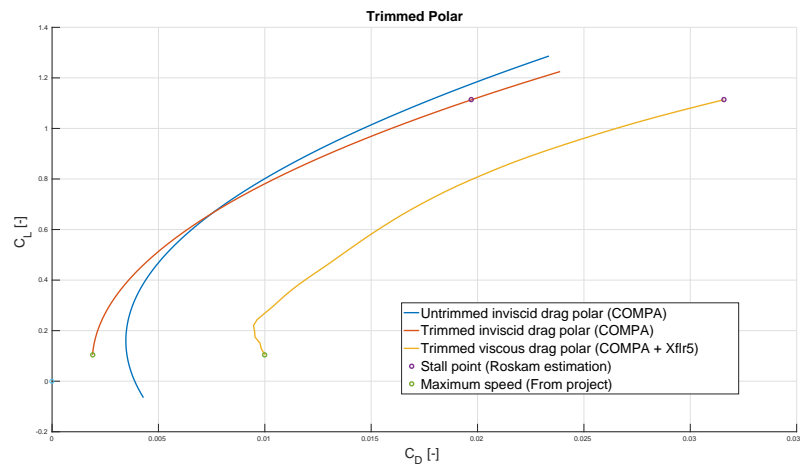


FIGURE 4.5: Trimmed polar.

# Chapter 5

# Xflr5 unsteady analysis

## 5.1 Mesh

The mesh for the Xflr5 VLM unsteady tests was designed with the software internal mesh tool. Two different meshes were designed to be compared, one featuring only the three lifting surfaces and one that includes a fuselage. Both meshes were validated manually to find the right number of panels. The meshes for the two geometries are equal except for the presence of the fuselage and the thickening of the mesh at the wing root in the full geometry case.

**Wing**    The wing is meshed with a cosine distribution along the chord and a uniform distribution along the wing span for all sections except the tip blocks which features a sine distribution. In the geometry that includes the fuselage the wing root is also thickened with a sine distribution. The final panel count is 6 in the chord direction and 84 in the span direction, for a total of 504 for the full geometry case and $6{\times}80$ for a total of 480 in the three surfaces case. Since a control derivative analysis is not needed with this mesh, the control surfaces were not included in the meshes.

**Horizontal empennage**    The horizontal empennage is meshed with a uniform distribution along the chord, a uniform distribution along the span of the root block and a sine distribution along the span of the tip block. The final panel count is 5 in the chord direction and 30 in the span direction, for a total of 150.
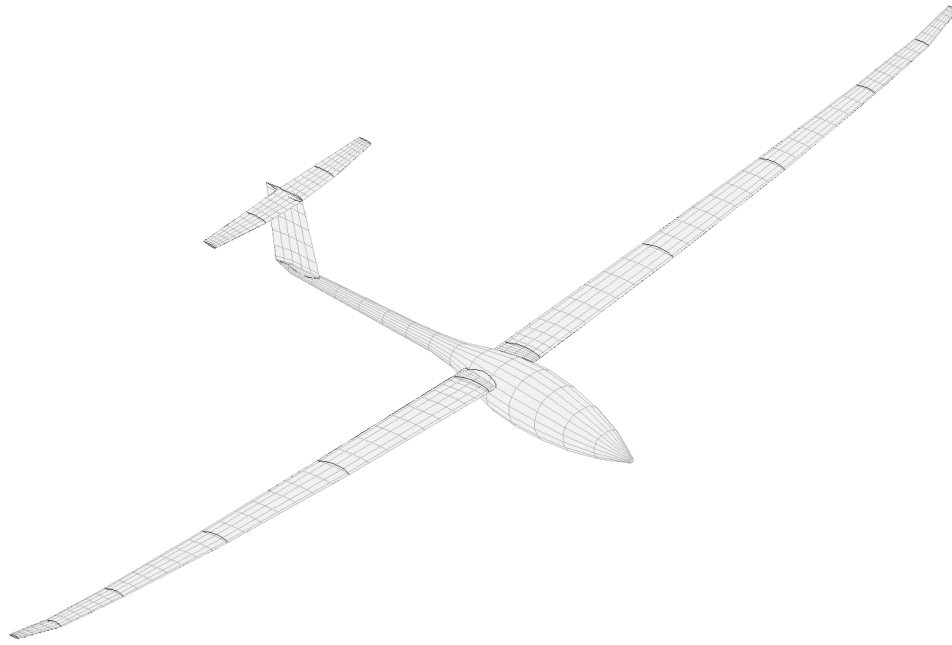
(a) Full aircraft mesh
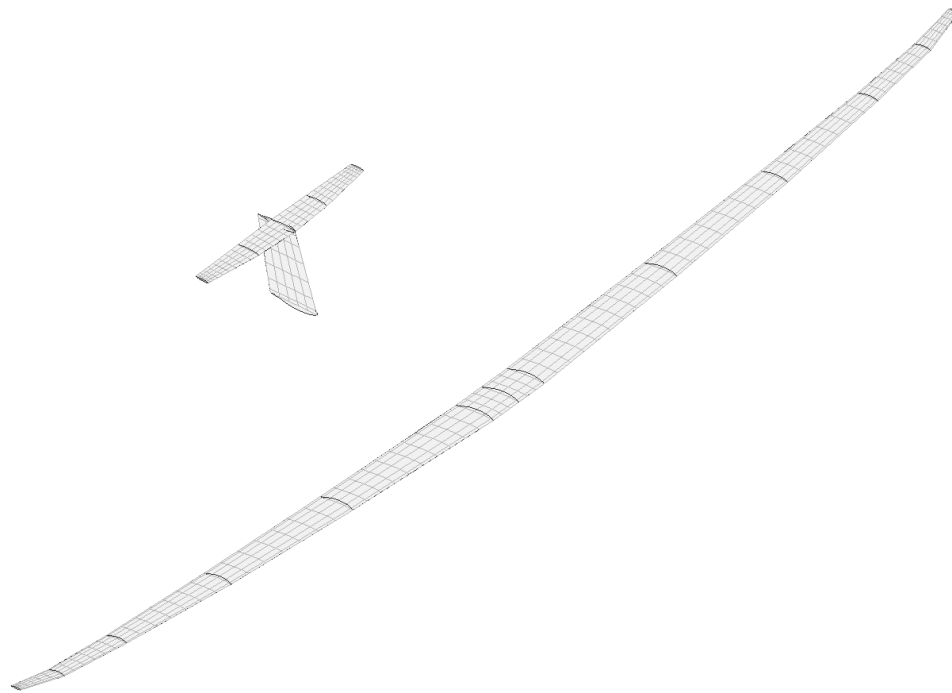


(b) Three surfaces mesh

FIGURE 5.1: Xflr5 VLM meshes.

**Vertical empennage** The vertical empennage is meshed with a uniform distribution along the chord and a cosine distribution along the span. The final panel count is 5 in the chord direction and 7 in the span direction, for a total of 35.

**Fuselage** The fuselage is a mesh with a uniform distribution of panels along its length and section. It features 20 panels in the section and 15 along the length, for a total panel count of 300.

The total panel count for the full geometry mesh is 989 (Figure 5.1(a)) and for the three surfaces mesh is 665 (Figure 5.1(b)). The VLM method needs substantially less panels than the 3D panels method in COMPA.

## 5.2 Steady-state tests

The two meshes were tested in a steady analysis at $\alpha = [-6°, -3°, 0°, 3°, 6]^T$. The aerodynamic curves are shown and compared with COMPA results in Figures 5.2,5.3,5.4 and 5.5. Figure 5.6 shows the pressure distribution and wake flow streamlines for the two geometries. Note that the streamlines are generated by an ad-hoc Xflr5 post-processor, the actual wake used by Xflr5 in this computation is straight, steady and directed as the asymptotic velocity.



FIGURE 5.2: Comparison between $C_L - \alpha$ curves of COMPA and of Xflr5 geometries with and without the fuselage.

FIGURE 5.3: Comparison between $C_D - \alpha$ curves of COMPA and of Xflr5 geometries with and without the fuselage.
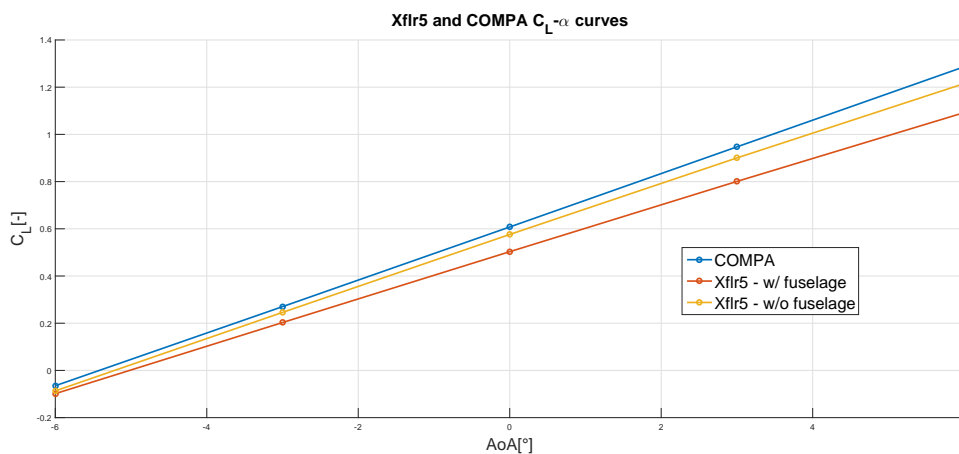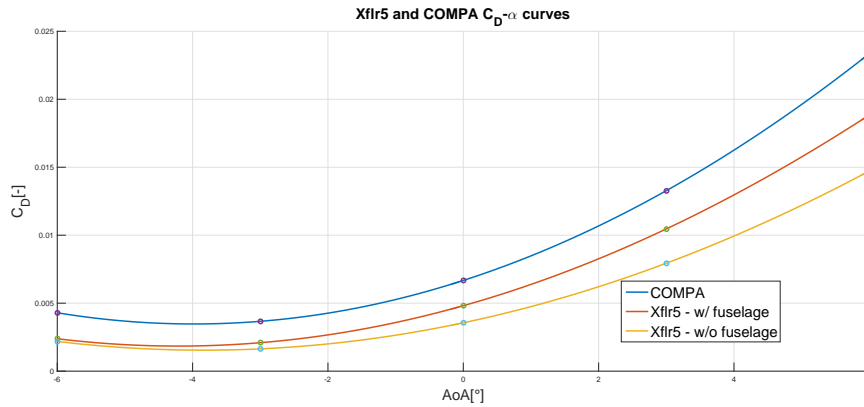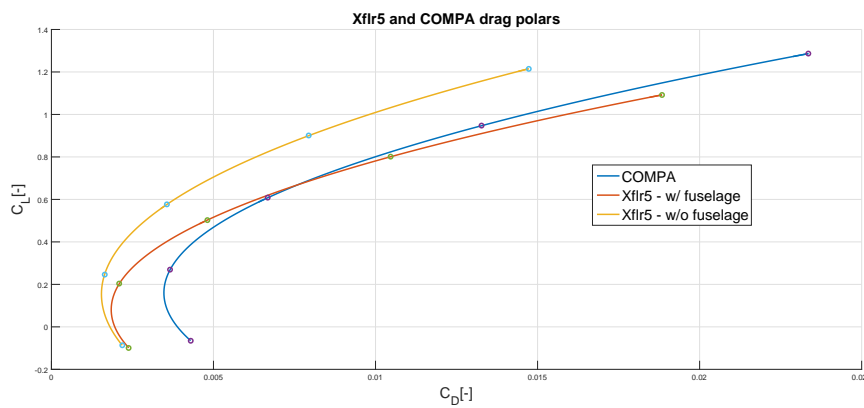


FIGURE 5.4: Comparison between the drag polar curves of COMPA and of Xflr5 geometries with and without the fuselage.
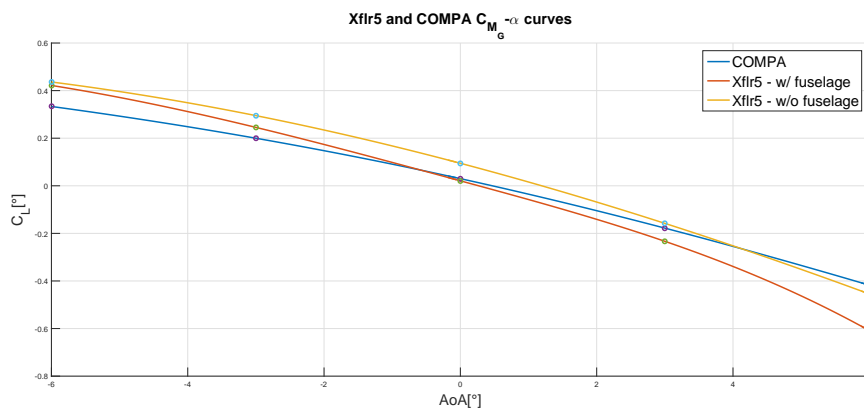


FIGURE 5.5: Comparison between the longitudinal moment curves of COMPA and of Xflr5 geometries with and without the fuselage.
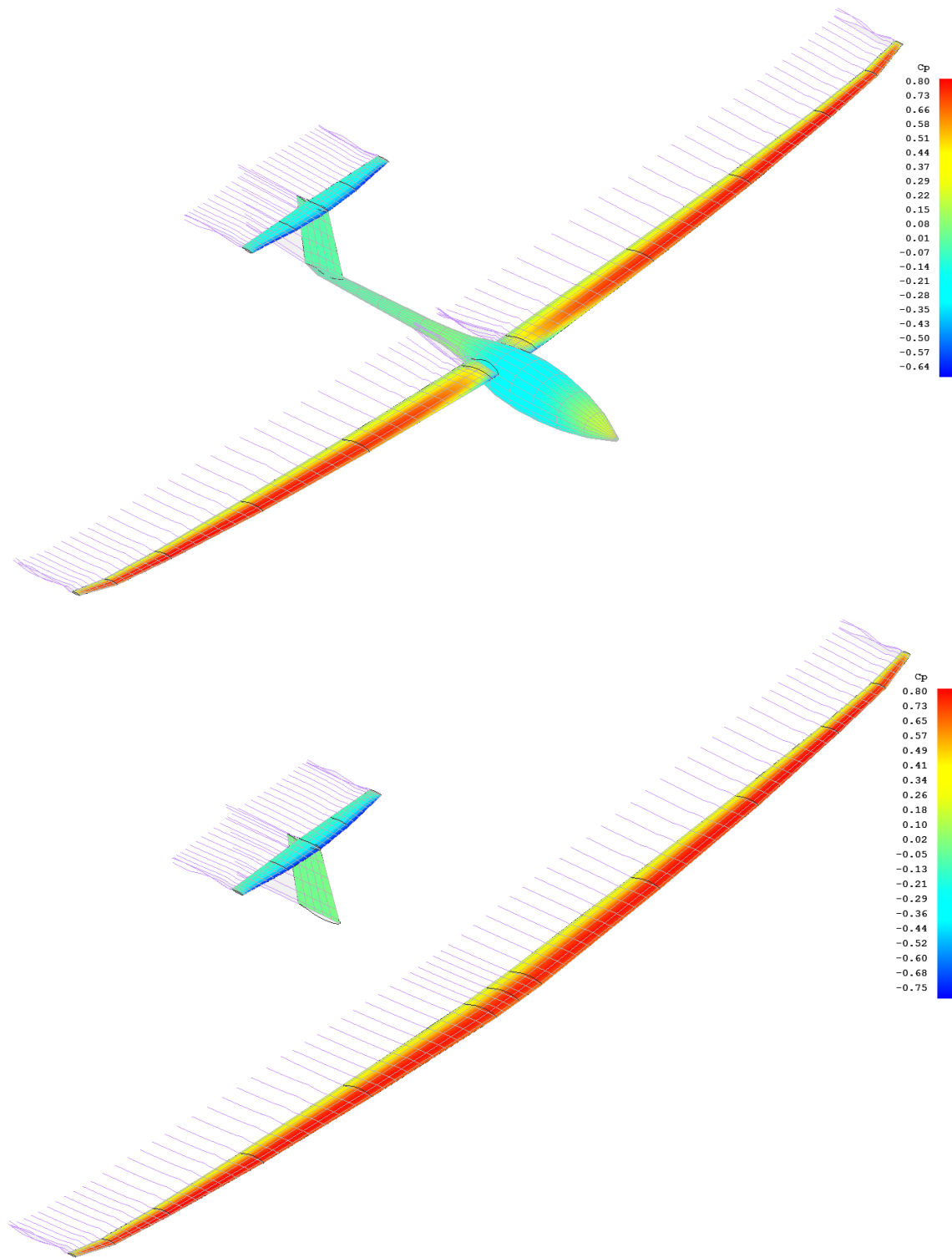
FIGURE 5.6: Resulting pressure distribution and wake
streamlines for the two Xflr5 geometries.

## 5.3 Unsteady analysis

The Unsteady Xflr5 analysis was computed on the 3 surfaces geometry, with the weight of a typical flight configuration (W=430 kg). This analysis is not intended to be as refined as the derivatives calculation with COMPA, like the viscous drag calculation, this is rather an estimation of the terms that are missing from the aerodynamic model.

Not having control surfaces, it is impossible to trim the aircraft for a given flight condition in a conventional way. The sole way to achieve equilibrium is to find the only couple of $\alpha$ and $v_e$ values that balances both the vertical force and longitudinal moment equilibrium equations. This happens at: $\alpha = 1.3°$ and $v_e = 28.27\text{m/s}$, this condition is not far from condition 1 of the COMPA analysis, which is used as a reference .The unsteady analysis is performed in this condition.

TABLE 5.1: Xflr5 longitudinal stability derivatives.

| | |
|---|---|
| $C_{x,\alpha}$ | 0.64 |
| $C_{L,\alpha}$ | 6.31 |
| $C_{L,q}$ | 12.63 |
| $C_{My,\alpha}$ | $-4.28$ |
| $C_{My,q}$ | $-55.828$ |

Xflr5 computed aerodynamic centre: $x_{A_{Xflr5}} = 3.14665$ m

TABLE 5.2: Xflr5 lateral stability derivatives.

| | |
|---|---|
| $C_{Y,\beta}$ | $-0.34$ |
| $C_{Y,p}$ | $-0.20$ |
| $C_{Y,r}$ | 0.21 |
| $C_{Mx,\beta}$ | $-0.24$ |
| $C_{Mx,p}$ | $-1.4$ |
| $C_{Mx,r}$ | 0.38 |
| $C_{Mz,\beta}$ | 0.12 |
| $C_{Mz,p}$ | $-0.18$ |
| $C_{Mz,r}$ | $-0.06$ |

The steady results from this analysis can be compared to those of the COMPA analysis in Tables 3.3, 3.4, 3.5 and 3.6

# Chapter 6

# Conclusion

## 6.1 Remarks

This thesis achieves its twofold purposes of studying the V5 "Rondone" aerodynamics and building a COMPA pre-processor capable of automatically building a mesh for a great variety of aircraft geometries. A complete mesh totalling 12440 panels has been obtained after a substantial process of development with dedicated geometry definition procedures for each category of components that are present in a complete airplane: lifting surfaces, fuselage, junctions, control surfaces. This mesh, after a thorough convergence study, has been used to estimate the aerodynamic characteristics of the glider in non-maneuvering flight conditions. Viscous and unsteady contributions that could not be obtained through the use of the COMPA aerodynamic solver were determined using available methods and tools in the public domain (Xflr5 VLM).

The main difficulties that were faced in this work were related to the understanding of the limits of available computational tools. Indeed, previous applications of the COMPA aerodynamic solver were limited to a few applications and did not engender a general framework capable for handling arbitrary aircraft geometries. COMPA proved to be a very reliable and consistent software after understanding the requirements applicable to the geometric grid, and such requirements took a long time to be thoroughly tested and assimilated. Also, the scarcity of data about sailplanes in the aeronautical literature made the task of estimating the viscous drag coefficients for the V5 quite consuming.

Xflr5 provides a very complete packages of methods and is very simple to use: possibly its best quality is the small time in which it is possible to set up completely new analysis. On the other hand, its own simplicity limits its potential, because it is not possible to customize the mesh point-by-point to optimize it as it is the case in COMPA. Also, the uncertainties on its results (except for the LLT viscous drag computation) show that Xflr5 really is: a preliminary design tool, used mostly by amateurs, with no ambition to be used professionally.

The main bulk of this work was the pre-processor development, which may be used in future works to study the aerodynamics of other aircrafts, or may be be embedded in a optimisation algorithm. The additional codes developed for mesh validation and aerodynamic derivative analysis are also applicable on the vast majority of aircraft configurations, so this project leaves behind a complete software package for the study of aircrafts.

## 6.2   Future developments

Future developments that include further work on the V5 "Rondone". In particular, a better estimation of the viscous drag using additional CFD tools and more computational power. Local aerodynamic optimization studies are also possible: the optimisation of the junctions to minimise interference drag, especially at the wing/fuselage interface, or an investigation on the advantages of mounting winglets. Some of these works would be nicely complemented by experimental wind tunnels tests.

From a computational point of view, a further development of COMPA, to include unsteady analyses, would be a great enhancement of this work.

# Bibliography

[1] J. Roskam. *Airplane design*. Vol. 1-7. DARcorporation, 1985.

[2] *DECISION NO. 2003/13/RM OF THE EXECUTIVE DIRECTOR OF THE AGENCY of 14 November 2003 on certification specifications, including airworthiness codes and acceptable means of compliance for sailplanes and powered sailplanes (« CS-22 »).* EASA, 2003.

[3] F. De Florio. *Airworthiness: an introduction to aircraft certification*. Elsevier, 2010.

[4] G. Droandi and G. Gibertini. "Assessment of a propeller model embedded on a panel method code for aircraft aerodynamics". In: *The Journal of Aerospace Science, Technology and Systems* 91.3/4 (2012), pp. 98–108.

[5] R.D. Finck and D.E. Hoak. *USAF stability and control DATCOM*. Engineering Documents, 1978.

[6] B.N. Pamadi and L. W. Taylor Jr. *Semiempirical method for prediction of aerodynamic forces and moments on a steadily spinning light airplane*. Tech. rep. NASA Langley Research Center, 1987.

[7] M. Enomoto and Y. Yamamoto. "Modelling, simulation and navigation experiments of Unmanned Aerial Vehicle". In: *Mechatronics and Automation (ICMA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 482–487.

[8] T. Melin. "A vortex lattice MATLAB implementation for linear aerodynamic wing applications". Royal Institute of Technology, 2000.

[9] A. Deperrois. *XFLR5 Analysis of foils and wings operating at low reynolds numbers, 2009*. 2010.

[10] Z. Li. "Modeling and Simulation of Autonomous Thermal Soaring with Horizon Simulation Framework". Cal Poly, San Luis Obispo, 2014.

[11] S.R. Rubio. "Aerodynamic analysis of a sustainable-energy sailplane using ceasiom". PhD thesis. Royal Institute of Technology, 2013.

[12] C. Sequeira, D. Willis, and J. Peraire. "Comparing aerodynamic models for numerical simulation of dynamics and control of aircraft". In: *44th AIAA Aerospace Sciences Meeting and Exhibit*. AIAA. 2006, p. 1254.

[13] P.A. Henne. *Applied computational aerodynamics*. Vol. 125. AIAA, 1990.

[14] J.C. Sivells and R.H. Neely. *NACA Technical Report 865 - Method for calculating wing characteristics by lifting-line theory using nonlinear section lift data*. Tech. rep. National Advisory Committee for Aeronautics, 1947.

[15] DP Coiro and F Nicolosi. "Design of low-speed aircraft by numerical and experimental techniques developed at DPA". In: *Aircraft Design* 4.1 (2001), pp. 1–18.

[16] M. Milani. "Aerodynamic shape optimization of tailless aircraft". Politecnico di Milano, 2016.

[17] B. Maskew and F.A. Woodward. "Symmetrical singularity model for lifting potential flow analysis". In: *Journal of Aircraft* 13.9 (1976), pp. 733–734.

[18] J. Roskam and C.T.E. Lan. *Airplane aerodynamics and performance*. DARcorporation, 1997.

[19] D.P. Raymer. *Aircraft design: a conceptual approach and Rds-student, software for aircraft design, sizing, and performance set (AIAA Education)*. AIAA & Ast, 2006.

[20] M. Sadraey. *Aircraft Performance: Analysis*. VDM Publishing, 2009.

[21] B. Etkin and L.D. Reid. *Dynamics of flight: stability and control*. Vol. 3. Wiley New York, 1996.

[22] B.N. Pamadi. *Performance, stability, dynamics, and control of airplanes*. Aiaa, 2004.

[23] W.F. Phillips, N.R. Alley, and R.J. Niewoehner. "Effects of Nonlinearities on Subsonic Aerodynamic Center". In: *Journal of Aircraft* 45.4 (2008), pp. 1244–1255.

[24] T. Hansen. "Modeling the Performance of the Standard Cirrus Glider using Navier-Stokes CFD". In: *Technical Soaring* 38.1 (2014), pp. 5–14.