



*People's Democratic Republic of Algeria Ministry of Higher
Education and Scientific Research*



MOSTEFA BENBOULAI D BATNA 2 UNIVERSITY

FACULTY OF TECHNOLOGY

Department of Mechanical Engineering

Final year Report

Automated semi-circular winglets design

Submitted by :

Chadia DJERIBIAI

Supervised by :

Dr : Laïd MESSAOUDI

DEDICATIONS

*In the Name of God, Most Gracious, Most Merciful
All the Pries is due to God alone, the Sustainers of
all the worlds*

*Thanks to the most graceful and compassionate, the Almighty
(Allah) who provided me with strength and patience to pass
through the exciting process of this research. I dedicate this
thesis and everything I do to my friend **Nadia BOUGUERN** for
being a source of encouragement and inspiration I wouldn't have
got this far without your support, to my parents to all the
members of my family, and for all my friends a special feeling of
gratitude for their support.*

ACKNOWLEDGMENTS

*This work would have never been possible without the help of the following people: first of all **Dr Laïd MESSAOUDI**, my supervisor, who was of great help to me, and whose constant patience and support helped a lot in the shaping of this research work during all its stages. Many thanks go to all my former and everlasting teachers during my university years who endeavour to bring their best, sharing their knowledge and expertise without moderation during all the academic years. Thanks for being such inspiring people. Thanks are also due to all the colleagues I enjoyed working and learning. Thanks a lot for your enthusiasm and support.*

ملخص

الغرض من هذا المشروع هو تسهيل نمذجة تصميم الجناح، وأربع أنواع مختلفة من الجنيحات من خلال جعل بنائها اليا. تم تصميم نموذج جناح الطائرة 200-757 بدون جنيح، ثم الأربع جنيحات (جنيح مائل، نصف دائري، الحلزوني، والوايت كومب) ورسمها في برنامج بلاندر، استخدام لغة البرمجة بايثون لإنشاء واجهة جرافيكية مشتركة لرسم الجناح والأربع جنيحات والتي تحتوي على أدوات لإدخال مختلف التعديلات على التصميم وفقا لشروط كل حالة.

Résumé

Le but de ce projet est de faciliter la modélisation de la conception d'aile, et quatre types différents de winglets en automatisant leurs conceptions. Le modèle d'aile d'avion 757-200, quatre winglets (trapézoïdale, semi-circulaire, spiroïdale, Whitecomb), ont été conçus et dessinés par Blender, en utilisant le langage de programmation Python, nous avons créé une interface graphique commune pour dessiner l'aile et les winglets.

ABSTRACT

The purpose of this project is to make the wing and winglet modeling easier by making the conception of them automated. 257-200 Aircraft wing model without winglet and four different type of winglets (Blended, Whitcomb, Semi-circular, and Spiroid winglet) were created and drawn in Blender, then use python programing language to create common graphical user interface to draw the wing and the four winglet type, which contain tools to enter such modification on the design, according to each case conditions.

Table of contents

Dedications

Acknowledgments

Resume

General Introduction.....5

Chapter I: USING WINGLET.....6

I.1 Introduction.....7

I.1.2 History7

I.1.3 History of Wingtip Devices and Winglets8

I.1.4 Added Winglets in Boeings Planes.....9

I.2 Aerodynamics.....9

I.2.1 Definition.....9

I.2.2 Airfoils.....9

I.2.3 Parameters of Naca 4-Digit Airfoil Sections.....10

I.2.4 Naca Type Profiles.....12

I.2.5 Four-Digit Series.....12

I.2.6 Five-Digit Series.....13

I.2.7 Shape of the Airfoil.....13

I.2.8 The Four Aerodynamic Forces.....14

I.3 Winglet.....17

I.3.1 How To Choose The Right Winglet?17

I.3.2 Measure Efficiency of Winglet.....18

I.3.1 Definition.....18

I.3.3 Comparison with Spin Efficiency Factor.....19

I.3.4 Different Types of Winglets.....20

I.3.5 The Benefits of Winglets.....21

<i>Conclusion</i>	22
<i>Chapter II: SEMI-CIRCULAR WINGLET</i>	23
<i>II.2 Description</i>	24
<i>II.3 Dimensions of Semi-Circular Winglet</i>	25
<i>II.4 Winglet Airofoil</i>	25
<i>II.5 Boeing 757-200's Wing</i>	26
<i>Chapter III: PROGRAMMING TOOLS</i>	28
<i>III.1 introduction</i>	29
<i>III.1.1 Open Source Software</i>	29
<i>III.1.2 Open Source Movement</i>	29
<i>III.1.3 Open Source Licenses</i>	30
<i>III.1.4 Stable Releases</i>	30
<i>III.1.5 Advantages of Open Source Software</i>	30
<i>III.2 Blender</i>	32
<i>III.2.1 What Is Blender? Executive Summary</i>	32
<i>III.3 Python</i>	32
<i>III.3.1 What Is Python? Executive Summary</i>	32
<i>III.4 Javafoil</i>	33
<i>III.4 Conclusion</i>	34
<i>CHAPTER IV: GUI REALISATION</i>	35
<i>IV.1 Preface</i>	36
<i>IV.2 User Interface</i>	36
<i>IV.3 Winglet and Wing Design Process</i>	40
<i>IV.3.1 Blender and Python</i>	41
<i>IV.3.1.1 Introduction</i>	41
<i>IV.3.1.2 The bpy Module</i>	41

<i>IV.3.1.4 Running Scripts.....</i>	<i>45</i>
<i>IV.3.1.5 Python Console</i>	<i>45</i>
<i>IV.3.2 Generate the Template of an Airfoil of Desired Coordinate Points.....</i>	<i>46</i>
<i>IV.3.2.1 Javafoil.....</i>	<i>46</i>
<i>IV.3.3 Running Our Script.....</i>	<i>50</i>
<i>IV.3.3.1 importing One Profile</i>	<i>50</i>
<i>IV.3.3.2 Profiles Realization.....</i>	<i>50</i>
<i>IV.3.3.3 Creating Panels.....</i>	<i>51</i>
<i>IV.3.3 Profiles Positions.....</i>	<i>53</i>
<i>IV.3.4 Three dimensional drawing of the wing.....</i>	<i>55</i>
<i>IV.3.5 Wing Exportation to Stl.....</i>	<i>57</i>
<i>IV.4 Three dimensional drawing of the Winglet.....</i>	<i>57</i>
<i>General Conclusion.....</i>	<i>60</i>

General Introduction

Aviation has become the main passenger transportation method worldwide. Due to this fact, it is considered a pivotal economic activity for all countries. As the case with every major economic activity, it is desirable to make it more widespread and affordable for more people. Therefore, engineering methods come to play in this regard. From the beginning of aviation, engineering has been the backbone of its development. Today, the field of aviation is too complicated. It has multiple facets ranging from international relations to simple office issues. Airline business could be the most diverse type of business, as the company has to deal with diverse issues in order to run its operations successfully. Due to the rapid expansion and growing complexity, engineers are called upon to play new roles in the airline, in addition to their traditional roles. Universities worldwide have the responsibility to train such engineers to fulfill the needs of the aviation industry.

The number of planes in the air is expected to double over the next ten years. To cope with this phenomenon, the industry must redouble its efforts, particularly in terms of emissions, and economizing, in order to build less polluting aircraft with lack consumption of fuel. Today it is turning to areas such as cleaner fuels, lighter planes, better design, more efficient engines and better air traffic management. All of these initiatives will help meet the emission reduction targets set by industry and regulatory authorities and economize more.

Due to, the background we had as an aeronautics student with years of research and learning experience, we discovered through this stage, that Aircraft designers agreed that, the winglet is good way to solve the problem of pollution and fuel consumption, and its design play important role in the industry, therefor, this study involves build Automated winglets, by creating conception Addon interface of Boeing 757-200 aircraft wing and four different winglet with possibility to modify in their dimensions, using the open source software Blender for the conception and python language to run the program script.

The made interface cover both the academic and the professional dimensions, the academic dimension ensures that is scientific and pedagogical enough to be used for teaching or doing academic experiences by facilize the conception progress, while the

professional dimension ensures that this Blender Add-on interface is fit as a designing tool for practicing engineers. This thesis is structured as follows:

The first chapter in the thesis starts by detailing the history of Winglets and explaining the theoretical side of the subject then giving in the second chapter all the information about the aircraft 757-200, wing and winglet dimensions. The third chapter provides necessary information about the used tools that helped us to create the addon interface. Finally, the last chapter is devoted to the design of the winglets as well as the explanation of the result which is the practical interface that could be used in the industry.

Chapter I

USING WINGLETS

I.1 Introduction

At this initial stage, we will detail the history of Winglets as well as to consider their importance in the aeronautical industry.

Finally, this chapter will consider to attempt seeing some brief notions on aerodynamics.

I.1.2 History

The flying birds in the sky make human to dream flight. After centuries of research's and developments that dream of flight came true. Still human are learning from birds about flight. Today the history of aviation is spacious extended from a simple kite to supersonic aircrafts. The sector of aviation is so vast, which is always cultivating and reaching many milestones for better performance.

The invention of aircraft begins in 16th, 17th and 18th centuries. Lots of researches been conducted, theories developed, real life testes and ended up with some successful gliders. The start of 19th century change the aviation history. In aircraft many advancements made in aerodynamics, instruments, flight controls, etc. that leads today advanced efficient aircrafts.

From early days wing is the most fundamental part of aircraft structure. This the part which generate the lift force and this force carry the load of aircraft. Designing of the aircrafts always involves advance optimization of wing, which gives efficient aircraft structure.

Today the aviation industry faces another issue, which is the consummation of fuel.

In an era of rising fuel prices and environmental concerns, efforts to decrease fuel consumption and lower emissions are of profound interest to the aviation industry. In recent years there has been intense research and study into aerodynamic devices that provide commercial aircraft with longer range and more efficient rates of fuel consumption.

Since the 1970s, when the price of aviation fuel began spiraling upward, airlines and aircraft manufactures have looked at many ways to improve the operating efficiency of their aircraft. Winglets have become one of the industry's most visible fuel saving technologies and their use continues to expand. Moreover, they can reduce fuel consumption about 4% in many flights when the distance to crossing is more than 1800 km [2].The concept of winglets was born with a British aerodynamicist in the

late 1800s, but the idea remained on the drawing board until revived in the early 1970s by Dr Richard Whitcomb when the price of aviation fuel started spiraling upwards. Whitcomb a noted aeronautical engineer at NASA's Langley Research Center, refined the concept of the quill with wind tunnel testing and computer studies. He then predicted that full-size transport planes with winglets would achieve an improvement in cruise efficiency of between 6% and 9%.

A penne flight test program at the NASA Dryden Flight Research Center in 1979-80 validated the Whitcomb research when the test aircraft - a military version of the Boeing 707 airliner - recorded a rate in fuel consumption increased by 6.5%.

The winglet test program was conducted at Dryden in 1979-80, followed by several years of wind tunnel testing and analytical studies by Dr. Richard Whitcomb at NASA Langley. Whitcomb had studied the original winglet concept developed by British aerodynamicist F.W. Lancaster in the late 1800s. Lancaster's patented concept to a vertical surface at the wing tip would reduce drag. Whitcomb took this concept by making the vertical surface as a refined airfoil that reduces drag by interacting with the air circulation and vortex at the end of the wing.

I.1.3 History of Wingtip devices and Winglets

Endplate theory was the first to propose wingtip device and was patented by Fredrick W. Lanchester, British Aerodynamicist in 1897. Unfortunately, his theory could not reduce the overall drag of aircraft despite reducing the induced drag. The increase in the viscous drag during cruise conditions outruns the reduction in induced drag.

In July 1976, Dr. Whitcomb made a research at NASA Langley research center and developed the concept of winglet technology. According to Whitcomb, winglet could be described as the small wing like vertical structures which extends from the wingtip, aiming at reduction in induced drag when compared to other wing tip devices or extensions. He also claimed in his research that the winglet shows 20% reduction in induced drag when compared to tip extension and also improved lift-to-drag ratio [26]. In 1994 Aviation Partners Inc. (API) developed an advance design of winglet called blended winglet. Louis B. Gratzler from Seattle has the patent for blended winglet and intention of the winglet is to reduce the interference drag due to sharp edges as seen in the Whitcomb's winglet Also, Gratzler has the patent for the invention of spiroid-tipped wing in April 7, 1992 [26].

The main purpose of all the above inventions was to decrease the strength of wake vortex and to reduce induced drag [26].

I.1.4 Added winglets in Boeings planes

The Boeing Aviation Partner Inc. has created winglets that reduce fuel consumption by around 5%. They were available for the 757-200 and as an addition to the 1030 aircraft already available. The mechanism for incorporating winglets has been called retrofitting. The winglets were made with the aim of not only reducing fuel consumption, but more importantly, they were intended to reduce induced drag.

The winglets were registered by Boeing Aviation Partner Company at the U. S patent office on September 20, 1994 with patent number 53482531. These winglets have brought more benefits to airplanes [1].

I.2 Aerodynamics

I.2.1 Definition

Aerodynamics as the science describing body's movement in an air. Thus it is a branch of dynamics which deals with the motion of air and other gases, with the forces acting upon an object in motion through the air, or with an object which is stationary in a current of air. In effect, in aviation aerodynamics is concerned with three distinct parts. These parts may be defined as the aircraft, the relative wind, and the atmosphere[4].

I.2.2 Airfoils

An airfoil is a surface designed to obtain a desirable reaction from the air through which it moves. Thus, we can say that any part of the aircraft which converts air resistance into a force useful for flight is an airfoil. The blades of a propeller are so designed that when they rotate, their shape and position cause a higher pressure to be built up behind them than in front of them so that they will pull the aircraft forward. The model of a wing (Fig. I-1) gives an excellent example of streamlines around airfoil.

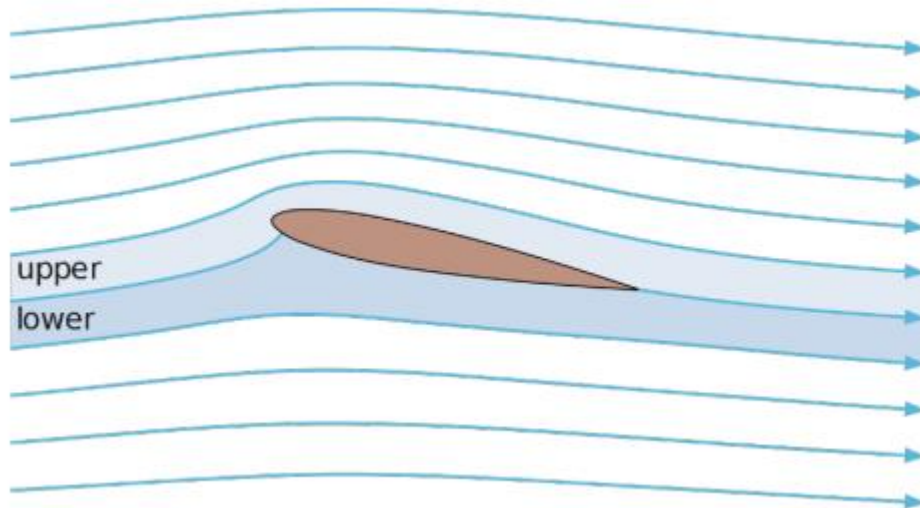


Figure I.1-Streamlines around airfoil [4].

Although the top surface of the conventional wing profile has greater curvature than the lower surface, the principal thing is the larger density of streamlines above the wing. The larger density of streamlines means the greater velocity of air.

According to Bernoulli's principle,

$$\frac{p}{\rho} + \frac{1}{2}q^2 + gz = C^t e \quad (I,1)$$

An increase in the speed of the fluid occurs simultaneously with a decrease in pressure or a decrease in the fluid's potential energy. This is equivalent to the principle of conservation of energy. This states that in a steady flow the sum of all forms of mechanical energy in a fluid along a streamline is the same at all points on that streamline[4].

I.2.3 Parameters of NACA 4-digit airfoil sections

We define an aerodynamic profile as the geometric contour obtained by the vertical section of an airplane wing. The shape of the profile is an essential characteristic of a wing and have great influence on the aerodynamic characteristics and performance of the aircraft. A profile is considered a two-dimensional wing with a wingspan that tends to infinity .

A geometric definition of the wing profiles is shown in figure (I.1)

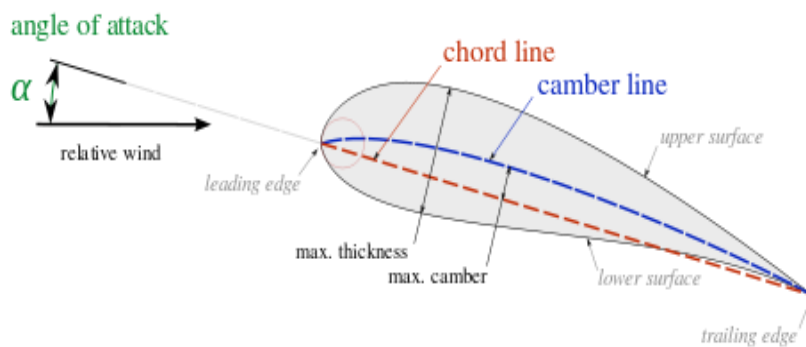
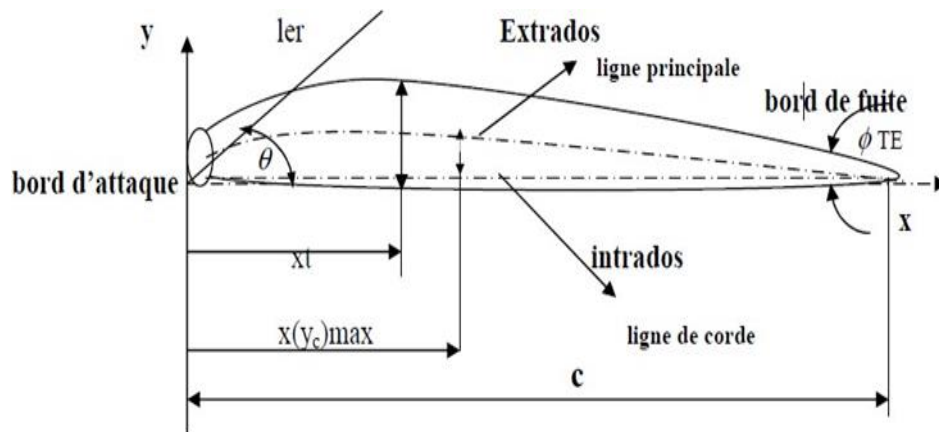


Figure 1.2 -Aerofoil Geometry[4].

The most important geometric parameters that define the aerodynamic characteristics of a profile are:

The maximum thickness, the ratio $\frac{e}{c}$

The shape of the camber.

The shape of the leading edge or the parameter ΔY

The angle of the trailing edge ϕ_{TE} (the arrow).

According to Figure 1.1, the geometric quantities of an aerodynamic profile are defined.

C: Chord of the profile section

x: Distance along the chord measured from the leading edge

Y: Ordinate with respect to several values of X (measured normal to X and from the chord line for symmetrical profiles after, and normal to X and from the mean line for arched profiles)

Y (X): distribution of the thickness of the profile

e = 2 : Y:max maximum thickness of the profile **l.er:** radius of the leading edge

ϕ_{TE} : the arrow (angle included between the tangents to the surfaces of the leading edge and the trailing edge).

1st: The mean line of the camber

(Yc) max: maximum ordinate of the mean line.

Yc (x): shape of the mean line

X (Yc) max: position of the maximum camber

θ : slope from the leading edge to the trailing edge, equal to the slope between the leading edge and the mean line.

ΔY : parameter of the shape of the nose.

The most recognized airplane wing profiles in the world are the famous American NACA type profiles and some of them are commonly used both in practice for the validation of numerical methods [1].

I.2.4 NACA type profiles

The NACA airfoils are airfoil shapes for aircraft wings developed by the National Advisory Committee for Aeronautics (NACA). The shape of the NACA airfoils is described using a series of digits following the word "NACA". The parameters in the numerical code can be entered into equations to precisely generate the cross-section of the airfoil and calculate its properties[3]

I.2.5 Four-digit series

The NACA four-digit wing sections define the profile by

- 1) First digit describing maximum camber as percentage of the chord.
- 2) Second digit describing the distance of maximum camber from the airfoil leading edge in tenths of the chord.
- 3) Last two digits describing maximum thickness of the airfoil as percent of the chord[3].

For example, the NACA 2412 airfoil has a maximum camber of 2% located 40% (0.4 chords) from the leading edge with a maximum thickness of 12% of the chord.

The NACA 0015 airfoil is symmetrical, the 00 indicating that it has no camber.

The 15 indicates that the airfoil has a 15% thickness to chord length ratio: it is 15% as thick as it is long.

I.2.6 Five-digit series

The NACA five-digit series describes more complex airfoil shapes. Its format is LPSTT,

where:

L: a single digit representing the theoretical optimal lift coefficient at ideal angle of attack $CL_i = 0.15 L$ (this is not the same as the lift coefficient CL),

P: a single digit for the x coordinate of the point of maximum camber (max. camber at $x = 0.05 P$).

S: a single digit indicating whether the camber is simple ($S = 0$) or reflex ($S = 1$),

TT: the maximum thickness in percent of chord, as in a four-digit NACA airfoil code.

For example, the NACA 23112 profile describes an airfoil with design lift coefficient of 0.3 (0.15×2), the point of maximum camber located at 15% chord (5×3), reflex camber (1), and maximum thickness of 12% of chord length (12).

Note: In our project we used the four-digit profile (NACA 2412) [3].

I.2.7 Shape of the Airfoil

The shape of a wing (Fig. I.3) consequently affects the efficiency of the wing.

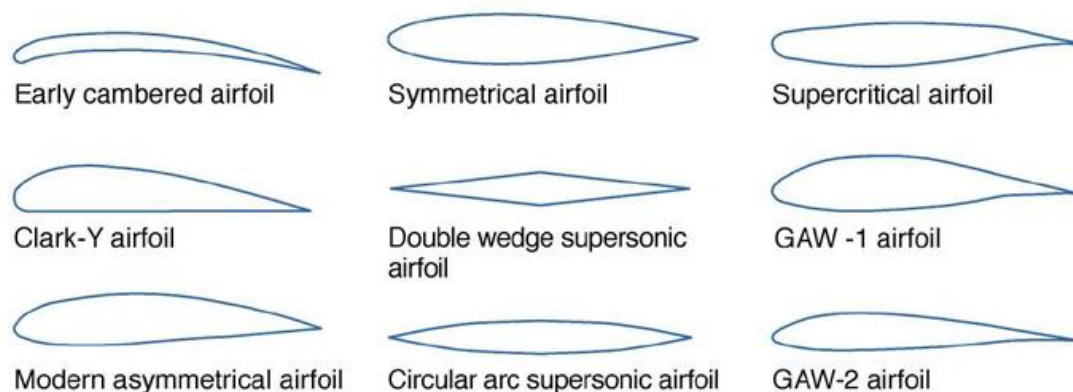


Figure I.3-Airfoil types [4].

Airfoil section properties differ from wing or aircraft properties because of the effect of the wing planform. A wing may have various airfoil sections from root to tip, with taper, twist, and sweepback.

The resulting aerodynamic properties of the wing are determined by the action of each section along the span.

Efficiency of a wing is measured in terms of the lift over drag (L/D) ratio. This ratio

varies with the angle of attack but reaches a definite maximum value for a particular angle of attack. At this angle, the wing has reached its maximum efficiency. The shape of the airfoil is the factor which determines the angle of attack at which the wing is most efficient; it also determines the degree of efficiency.

Research has shown that the most efficient airfoils for general use have the maximum thickness occurring about one-third of the way back from the leading edge of the wing.

High-lift wings and high-lift devices for wings have been developed by shaping the airfoils to produce the desired effect. The amount of lift produced by an airfoil will increase with an increase in wing chamber. Camber refers to the curvature of an airfoil above and below the chord line surface. Upper chamber refers to the upper surface, lower camber to the lower surface, and mean camber to the mean line of the section. Camber is positive when departure from the chord line is outward, and negatives when it is inward. Thus, high-lift wings have a large positive camber on the upper surface and a slight negative camber on the lower surface. Wing flaps cause an ordinary wing to approximate this same condition by increasing the upper chamber and by creating a negative lower chamber.

It is also known that the larger the wingspan as compared to the chord, the greater the lift obtained.

This comparison is called aspect ratio. The higher the aspect ratio, the greater the lift. In spite of the benefits from an increase in aspect ratio, it was found that definite limitations were of structural and drag considerations[3].

I.2.8 The four aerodynamic forces

There are four forces acting on the airplane all the time during airplane is flying. The four forces are (1) Lift, (2) Gravity force or Weight, (3) Thrust, and (4) Drag. Lift and Drag are considered aerodynamics forces because they exist due to the movement of the Airplane through the air.



Figure I.5- The four forces [3].

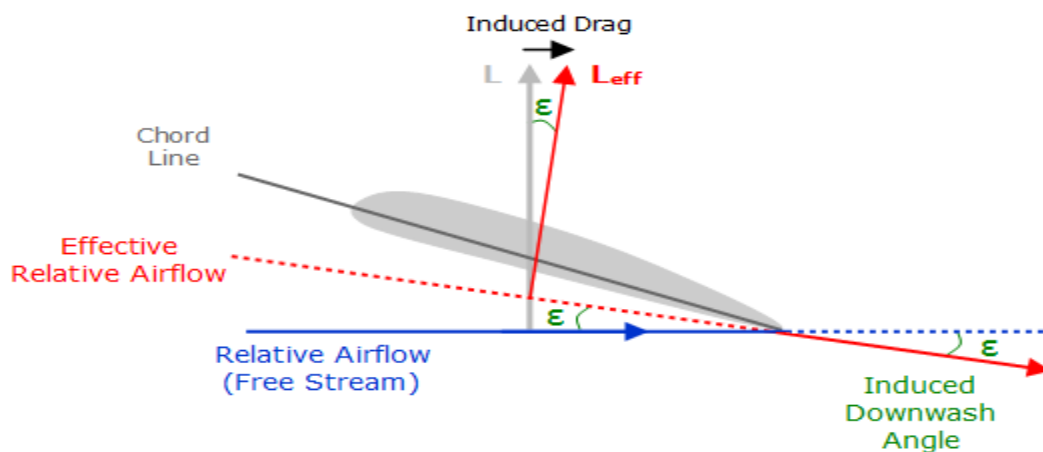


Figure I.6-Induce drag downwash[3].

Lift is produced by a lower pressure created on the upper surface of an airplane's wings compared to the pressure on the wing's lower surfaces, causing the wing to be lifted upward.

The special shape of the airplane wing (airfoil) is designed so that air flowing over it will have to travel a greater distance and faster resulting in a lower pressure area (see illustration) thus lifting the wing upward. Lift is that force which opposes the force of gravity (or weight)[11].

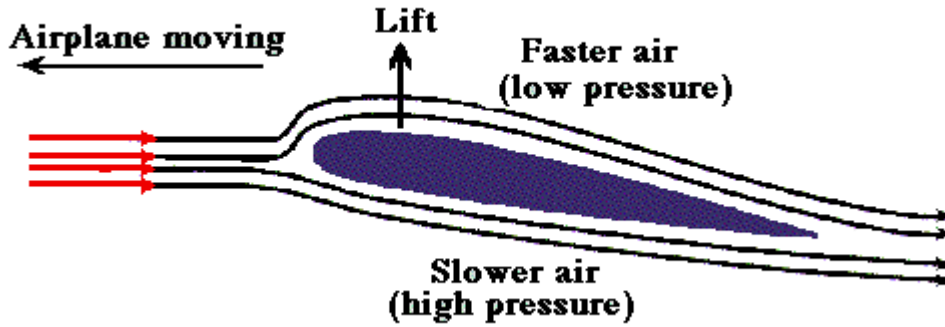


Figure I.7-Lifting [11].

Lift depends upon (1) shape of the airfoil (2) the angle of attack (3) the area of the surface exposed to the airstream (4) the square of the air speed (5) the air density.

✓ Lift equation

$$L = C_L \times d \times \frac{v^2}{2} \times A \quad (I.2)$$

L= Lift

d= density of air

C_L = lift coefficient

v = velocity of air

A = wing area

Weight: The weight acts vertically downward from the center of gravity (CG) of the airplane.

Thrust: is defined as the forward direction pushing or pulling force developed by aircraft engine . This includes reciprocating engines , turbojet engines, turboprop engines.

✓ Thrust equation

Thrust is a force, force is the change in momentum with time

$$F = \frac{[MV]_e - [MV]_0}{t_e - t_0} \quad (I.3)$$

m' = mass flow rate = mass/time

$$m' = d \times V \times A = \text{Density} \times \text{Velocity} \times \text{Area} \quad (I.4)$$

Thus $F = m'_e v_e - m'_0 v_0 \quad (I.5)$

Drag is the force which opposes the forward motion of airplane. specifically, drag is a retarding force acting upon a body in motion through a fluid, parallel to the direction of motion of a body. It is the friction of the air as it meets and passes over an airplane and its components. Drag is created by air impact force, skin friction, and displacement of the air

Drug equation [11].

$$D = C_D \times d \times \frac{v^2}{2} \times A \quad (1.6)$$

D= Drag

d = Density of Air

C_D = Drag coefficient

v= Velocity of Air

A = Wing Area

I.3 Winglet

Winglets, which are airfoils operating just like a sailboat tacking upwind, produce a forward thrust inside the circulation field of the vortices and reduce their strength. Weaker vortices mean less drag at the wingtips and lift is restored. Improved wing efficiency translates to more payload, reduced fuel consumption, and a longer cruising range that can allow an air carrier to expand routes and destinations. To produce as much forward thrust as possible, the winglet's airfoil is designed with the same attention as the airfoil of the wings themselves. Performance improvements generated by winglets, however, depend on factors such as the basic design of the aircraft, engine efficiency, and even the weather in which an aircraft is operating [4].

I.3.1 How to choose the right winglet?

Unfortunately, there is no magic formula for finding the "right" winglet. While it is certain that a large winglet will be more effective than a small one in reducing the induced drag, because it will more effectively oppose the formation of the marginal vortex while recovering more energy, it is not necessary to forget that the friction drag

and the forces on the structures will increase with the size of the winglet. As a case in point, let's look at the situation in which, the latter is optimized for certain flight conditions, in general the flight at high C_z when climbing or at high altitude, it could by its friction drag seriously degrade performance for flight conditions where the induced drag. Is naturally lower, in particular at low C_z and high speed [3].



Figure I.8-Example of winglet [3].

I.3.2 Measure efficiency of winglet

I.3.2.1 Definition

The Oswald efficiency is defined for the cases where the overall coefficient of drag of the wing or airplane has a constant + quadratic dependence on the aircraft lift coefficient.

$$C_D = C_{D0} + \frac{(C_L)^2}{\pi e_0 AR} \quad (I.7)$$

Where:

C_D : is the overall drag coefficient,

C_{D0} : is the zero-lift drag coefficient,

C_L : is the aircraft lift coefficient,

Π : is the circumference-to-diameter ratio of a circle,

e_0 : is the Oswald efficiency number

AR : is the aspect ratio

For conventional fixed-wing aircraft with moderate aspect ratio and sweep, Oswald efficiency number with wing flaps retracted is typically between 0.7 and 0.85. At supersonic speeds, Oswald efficiency number decreases substantially. For example, at Mach 1.2 Oswald efficiency number is likely to be between 0.3 and 0.5 [15].

1.3.3 Comparison with Spin Efficiency Factor

It is frequently assumed that Oswald efficiency number is the same as the span efficiency factor which appears in Lifting-line theory, and in fact the same symbol e is typically used for both. But this assumes that the profile drag coefficient is independent of C_L , which is certainly not true in general. Assuming that the profile drag itself has a constant+quadratic dependence on C_L , an alternative drag coefficient breakdown can be given by:

$$C_D = C d_0 + C d^2 (C_L)^2 + \frac{(C_L)^2}{\pi e AR} \quad (I.8)$$

Where

Cd_0 : is the constant part of the profile drag coefficient,

Cd_2 : is the quadratic part of the profile drag coefficient,

e : is the span efficiency factor from inviscid theory, such as Lifting-line theory
Equating the two C_D expressions gives the relation between the Oswald efficiency number e_0 and the lifting-line span efficiency e .

$$C_{D0} = C d_0$$

$$\frac{1}{e_0} = \frac{1}{e} + \pi AR C d^2 \quad (I.9)$$

For the typical situation $Cd_2 > 0$, we have $e_0 < e$ [15].

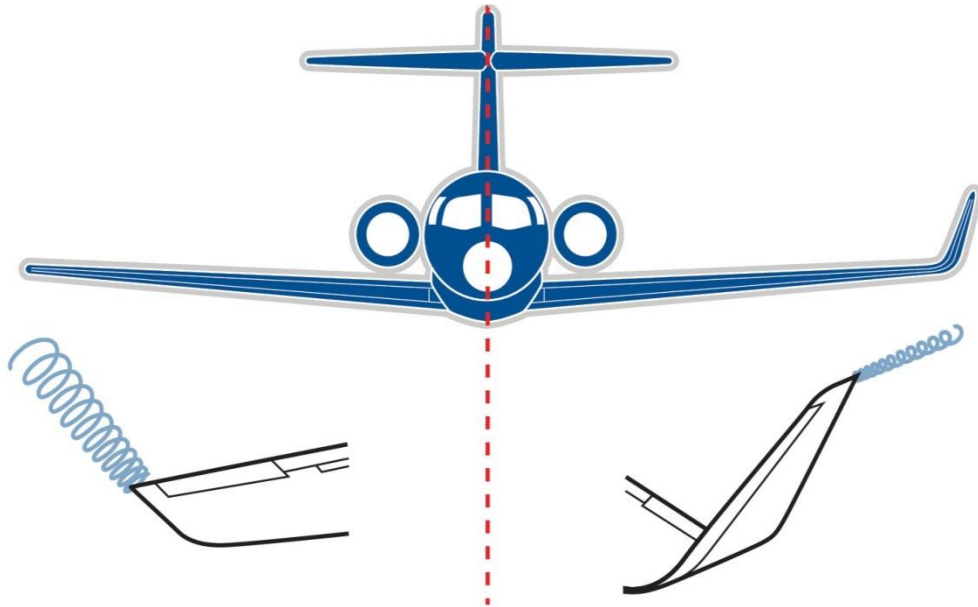


Figure I.9- Induced drag with and without winglet [4].

I.3.4 Different types of winglets

In general, any wingtips (Airline world) that not end the wing simply horizontally are considered as some kind of a winglet. Even though in strictly technical terms Wingtip Fences are not real extensions of the wing, and Raked Wingtips do not have a vertical part, they are still widely considered as winglet variants. We can distinguish three types of winglets: wingtip fences, blended winglets and raked wingtips. The different types of winglets are explained showing various images and commenting on the aircraft in which they are used [3].

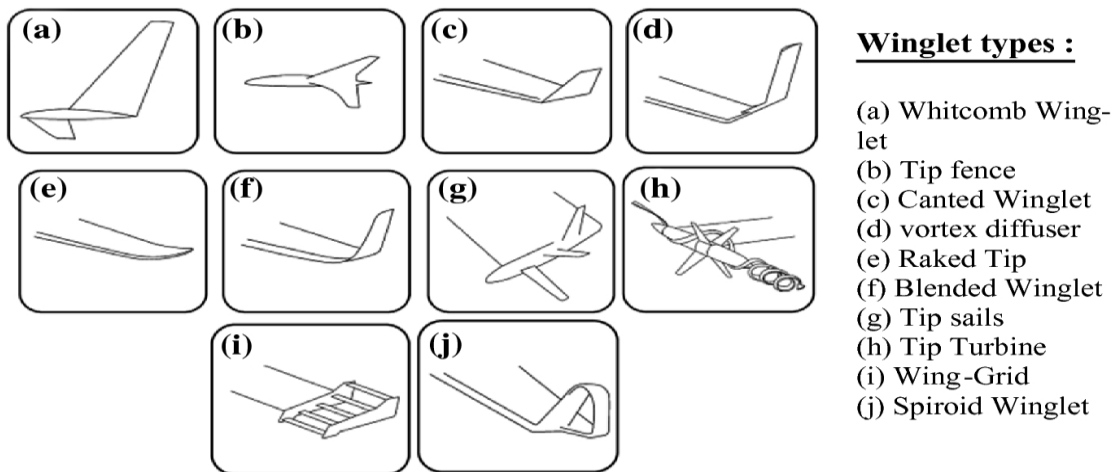


Figure I.10-Winglet types [1].

I.3.5 The Benefits of Winglets

Since the 1970s, when the price of aviation fuel began spiraling upward, airlines and aircraft manufacturers have looked at many ways to improve the operating efficiency of their aircraft. Winglets have become one of the industry's most visible fuel-saving technologies and their use continues to expand. Winglets increase an aircraft's operating efficiency by reducing what is called induced drag at the tips of the wings. An aircraft's wing is shaped to generate negative pressure on the upper surface and positive pressure on the lower surface as the aircraft moves forward. This unequal pressure creates lift across the upper surface and the aircraft is able to leave the ground and fly. Unequal pressure, however, also causes air at each wingtip to flow outward along the lower surface, around the tip, and inboard along the upper surface producing a whirlwind of air called a wingtip vortex. The effect of these vortices is increased drag and reduced lift that results in less flight efficiency and higher fuel costs. Winglets, which are airfoils operating just like a sailboat tacking upwind, produce a forward thrust inside the circulation field of the vortices and reduce their strength. Weaker vortices mean less drag at the wingtips and lift is restored. Improved wing efficiency translates to more payload, reduced fuel consumption, and a longer cruising range that can allow an air carrier to expand routes and destinations.

To produce as much forward thrust as possible, the winglet's airfoil is designed with the same attention as the airfoil of the wings themselves. Performance improvements generated by winglets, however, depend on factors such as the basic design of the aircraft, engine efficiency, and even the weather in which an aircraft is operating. The shapes and sizes of winglets, and the angles at which they are mounted with respect to the main wings, differ between the many types and sizes of aircraft produced but they all represent improved efficiency. Throughout the aviation industry, winglets are responsible for increased mileage rates of as much as 7%. Aircraft manufacturers and makers of add-on winglets have also reported improved cruising speeds, time-to-climb rates, and higher operating altitudes. The use of winglets throughout the aviation industry in the U.S. and overseas is constantly growing. Winglets now appear on powerless hang gliders soaring above mountain ridges and from seaside cliffs. Sailplane builders around the world have included blended winglets to their designs and the sleek, graceful gliders are silently soaring farther than ever Corporate-size Learjet's were the first commercial aircraft to

use winglets. Now, several decades later, winglets are incorporated into the designs of many other business jets such as Gulfstreams and the Global Express: a new aircraft built by Lear's parent company, Bombardier Retrofitting winglets to existing business jets is also a fast-growing market within the aviation industry itself. Many winglet marketing firms report their products help increase aircraft roll rates and lower approach and takeoff speeds [4].

I.3.6 Conclusion

The academic community has extensively explored that winglets are a great way to decrease induced drag by minimizing stress on the structure at the wing root. Further reduction of drag is essential to reduce consumption of fuel, therefor reduction of devices pollution. Indeed, much attention should be drawn to, the design of the connection between the wing and the winglet to avoid possible detachment phenomena which degrade performance. The next chapter will be devoted to the software used in the conception.

Chapter II

SEMI-CIRCULAR WINGLET

II.2 Description

Semi-circular winglet

It is a modern closed-loop wingtip device it is a relatively complicated design but easy to make since it is based on the spiral winglet shape and some modifications [10].

The semi-circular winglet profile, is the first design of spiroid winglet profile, is based on a varying aerofoil chord length as shown in picture below[27]



II.1-Semi-circular design [27].

For semi-circular winglets, one profile extends vertically the other one of the chord extends horizontally, and join in a spiral shape. The flow direction on the upper surface of a swept wing is normally from tip to root.

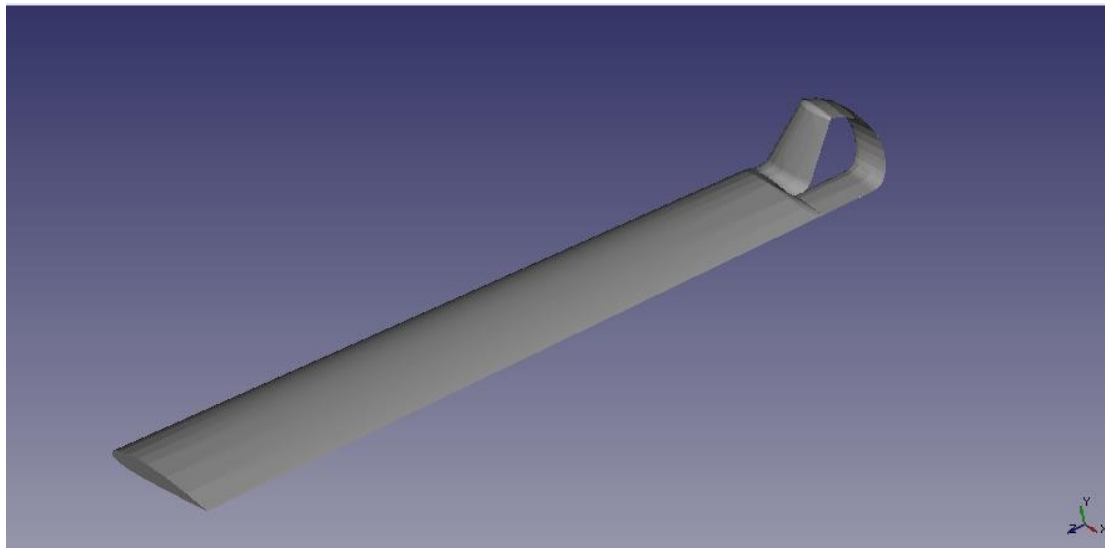


Figure II.1-Semi-circular winglet [18].

II.3 Dimensions of semi-circular Winglet

Due to the lack of information about this type of semi-circular winglet we customized it to fit the wing

✓ First profile

Extrude region= {"value":(0, 0, 0.6)}

Steps=30

Angle=3.1613

Center= (1, 1.3, 0.6)

Axis= (1, 0, 0)

✓ Second profile

Steps = 9

Angle=1.5708

Center= (1, 0.2, 0)

Axis= (1, 0.1, 0))

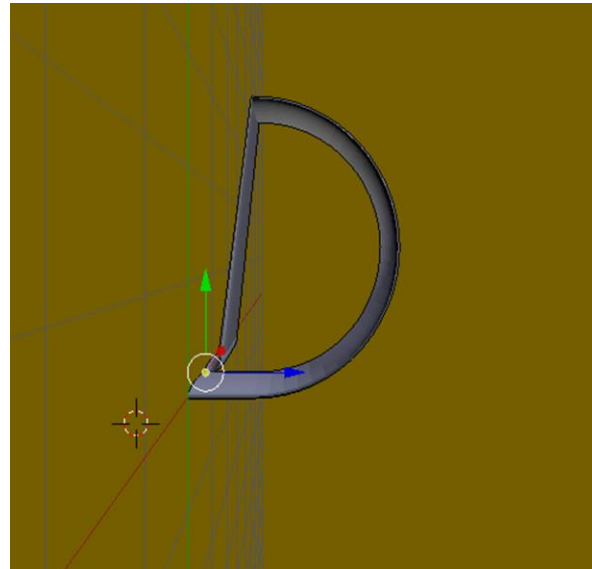


Figure II.2-Semi-circular Winglet.

II.4 Winglet Airofoil

The airofoil of winglet can be same as wing; also can be change according to requirement of lift and drag.

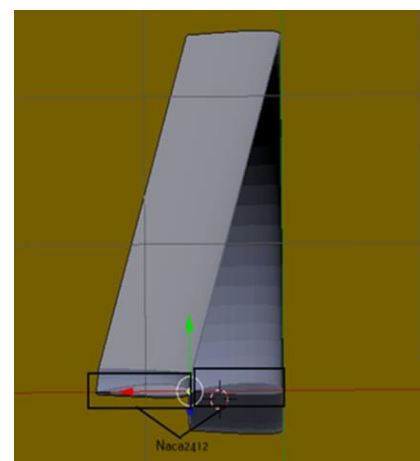


Figure II.3-Back view of the winglet.

II.5 Boeing 757-200's Wing

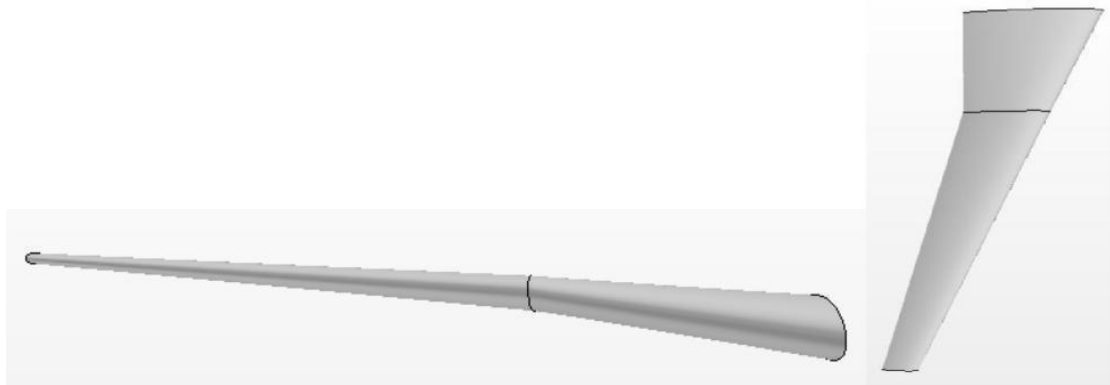


Figure II.4-3D Wing completed design[17].

- **Dimensions external**

wing chord at tip 1.73 Meter(5ft 8in)

wing aspect ration 7.8

length overall 47.33 meter (155 ft 3in)

length fuselage 46.97 meter (154ft 1 in)

tail plane span 15.22 meter (49 ft 11 in)

tail height 13.59 meter (min) (44 ft 6 in) / 13.74 meter (max) (45 ft 1 in)

wheel track 7.32 meter (24 ft)

wheel base 18.29 meter (60 ft)[28]

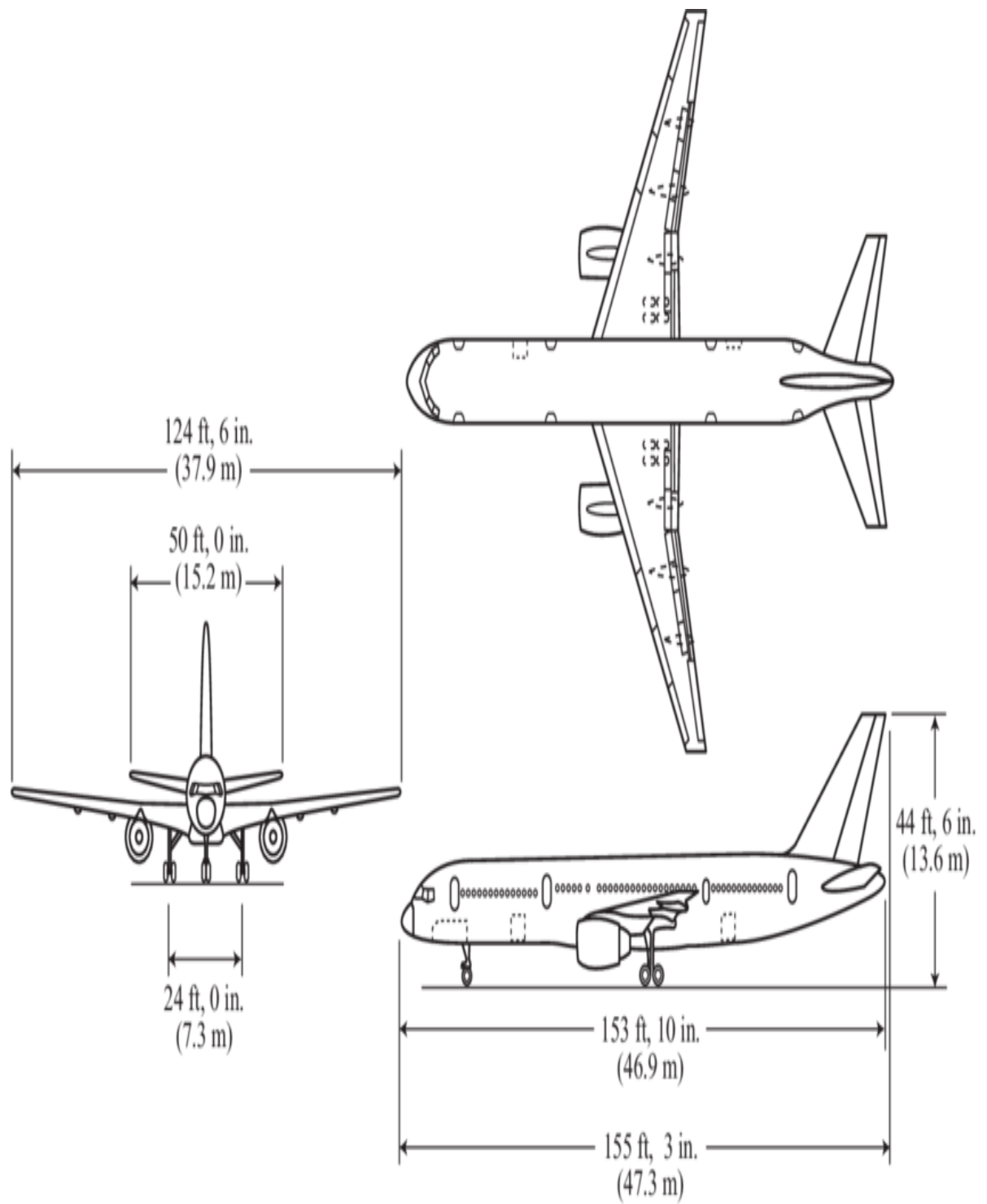


Figure II.5-NASA LaRC ARIES B-757-200 aircraft [29].

Chapter III

PROGRAMMING TOOLS

III.1 Introduction

Open source software is computer software that has a source code available to the general public for use as is or with modifications. This software typically does not require a license fee. There are open source software applications for a variety of different uses such as office automation, web design, content management, operating systems, and communications. In our project we have used open source software, therefore in this chapter we will provide you the necessary informations about the software we used on the modeling and on creation of our common winglet interface.

III.1.1 Open Source Software

III.1.1.1 Open Source Movement

In 1998, a group of individuals advocated that the term free software be replaced by open source software (OSS) as an expression which is less ambiguous and more comfortable for the corporate world. Software developers may want to publish their software with an open source software license, so that anybody may also develop the same software or understand how it works. Open source software generally allows anybody to make a new version of the software, port it to new operating systems and processor architectures, share it with others or market it. The aim of open source is to let the product be more understandable, modifiable, duplicatable, reliable or simply accessible, while it is still marketable. The Open Source Definition, notably, presents an open-source philosophy, and further defines a boundary on the usage, modification and redistribution of open-source software. Software licenses grant rights to users which would otherwise be prohibited by copyright. These include rights on usage, modification and redistribution. Several open-source software licenses have qualified within the boundary of the Open Source Definition. The most prominent example is the popular GNU General Public License (GPL). While open source presents a way to broadly make the sources of a product publicly accessible, the open-source licenses allow the authors to fine tune such access. The "open source" label came out of a strategy session held in Palo Alto in reaction to Netscape's January 1998 announcement of a source code release for Navigator (as Mozilla). A group of individuals at the session included Todd Anderson, Larry Augustin, John Hall, Sam Ockman, Christine Peterson and Eric S. Raymond. They used the opportunity before

the release of Navigator's source code to clarify a potential confusion caused by the ambiguity of the word "free" in English. The 'open source' movement is generally thought to have begun with this strategy session. Many people, nevertheless, claimed that the birth of the Internet, since 1969, started the open source movement, while others do not distinguish between open source and free software movements. The Free Software Foundation (FSF), started in 1985, intended the word 'free' to mean "free as in free speech" and not "free as in free beer." Since a great deal of free software already was (and still is) free of charge, such free software became associated with zero cost, which seemed anti-commercial [6].

III.1.1.2 Open source licenses

Open source licenses are assuring users freedom to use, copy, improve and distribution of software. GPL is the most popular license for free and open source software and provides feasible terms of use. Using GPL license, a user can modify the software without the permission of its creator. At the same time BSD license impose certain restrictions on modification of software without the permission of its developer. If you have decided to choose the software with non-General Public License, check the license if it contains any un-acceptable clauses [6].

III.1.1.3 Stable releases

Stable release of open source software shows its developers ability to fix and correct bugs along with new features. Version history of open source software is often available from project websites or any other project repositories like Source Forge (www.sourceforge.net), Savannah (savannah.net) and Free Software Foundations software directory (www.fsf.org). These services help users to check the information regarding software origin, releasing history, version numbering scheme, developers detail etc. Actively maintained open source projects mention even the releasing dates of forthcoming versions [6].

III.1.1.4 Advantages of Open Source Software

- **Lower Costs** Open source software usually does not require a licensing fee and its lower cost is generally one of the key reasons why small businesses

choose to adopt this software. Make sure that you consider the total costs of ownership when considering open source software.

- **Flexibility** A programmer can take a standard software package and modify it to better suit business needs. You can usually hire a programmer to add a particular function to open source software.
- **Reliability and Quality** When looking at improved quality, you have to compare the products themselves. It is impossible to say that open source software is better than proprietary software in terms of reliability and quality – both have a range of products. However, mature open source software is generally viewed to be of good quality and reliability. If your business is not familiar with open source software, you may only want to review some of the more mature products (e.g. Linux, Apache and Sendmail).
- **Reduces “Vendor Lock-in”** If you are using proprietary software you may be restricted to using certain vendors. Switching vendors in this case usually involves significant costs. Keep in mind though that choosing an OSS product may not make you totally independent of vendors. For some OSS products there may be a limited number of vendors that can provide you with services, upgrades or security patches.
- **Availability of External Support** External technical support is available for many of the open source software packages. Some vendors offer support contracts and there are service providers that install, configure and maintain an OSS system. Many open source products also have active online community support that may be able to answer your questions through online blog [7].
- **Simplified license management** Obtain the software once and install it as many times and in as many locations as you need. There’s no need to count, track, or monitor for license compliance [6].

You can find information about open source products on the Internet by using a search engine and typing the keywords “open source” followed by the “type of software application” that you are looking for now we will talk about "Blender", "Python" and "Java Foil" software as an example of open source software.

III.2 Blender

III.2.1 what is Blender? Executive Summary

People usually associate freeware software with the terms “bad”, “with limited features” or just a "demo". Blender is fully functional. It works as an open-sourced, community development program where people from around the world contribute to its success. Blender can be used to create 3D visualizations such as still images, 3D animations, modeling, VFX shots, and video editing. Blender is well suited to individuals and small studios who benefit from its unified pipeline and responsive development process. Blender is a cross-platform application, running on Linux, macOS, and Windows systems. Blender also has relatively small memory and drive requirements compared to other 3D creation suites. Its interface uses OpenGL to provide a consistent experience across all supported hardware and platforms [10].

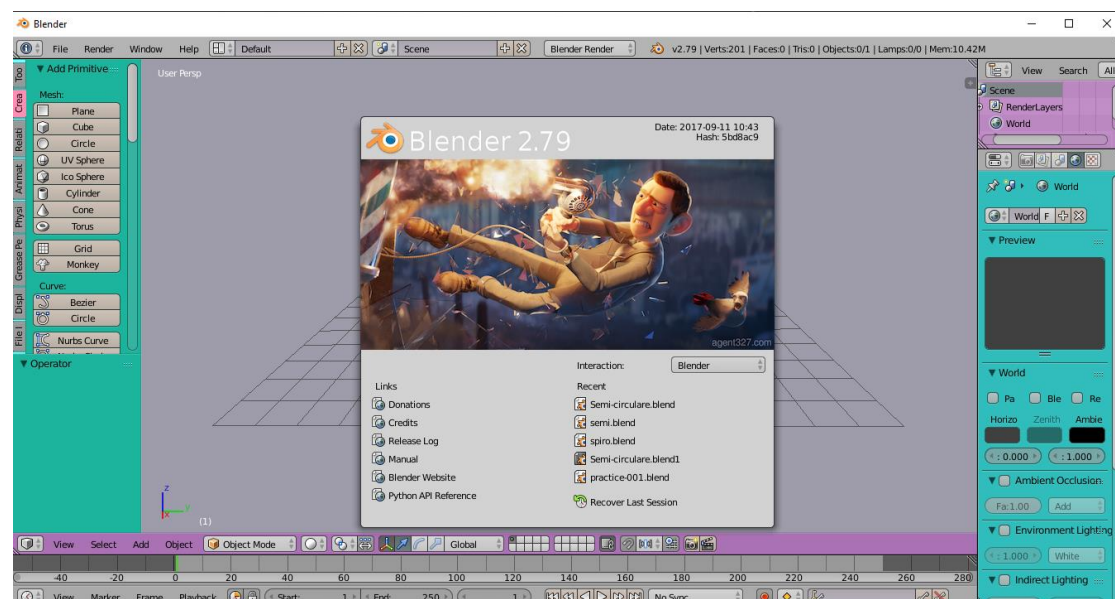


Figure III.1-Blender interface.

III.3 Python

III.3.1 What is Python? Executive Summary

The Python programming language was created in 1989 by Guido van Rossum, in the Netherlands. The name Python comes from a tribute to the television series Monty Python's Flying Circus of which G. van Rossum is a fan. The first public version of

this language was published in 1991. The latest version of Python is version 3, specifically, version 3.8. Python version 2 is now deprecated and it been cease supported since January 1, 2020. When possible, avoid use it. The Python Software Foundation¹ is the association that organizes the development of Python and leads the community of developers and users. This programming language has many interesting features [14]. python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed [13].

III.4 JavaFoil

Javafoil is a free program that allows you to perform wing analysis. The program takes a set of coordinate profiles; it calculates the local, viscous flow velocity along the wing surface for any angle of attack. It also solves a set of differential equations to find the different parameters of the boundary layer.

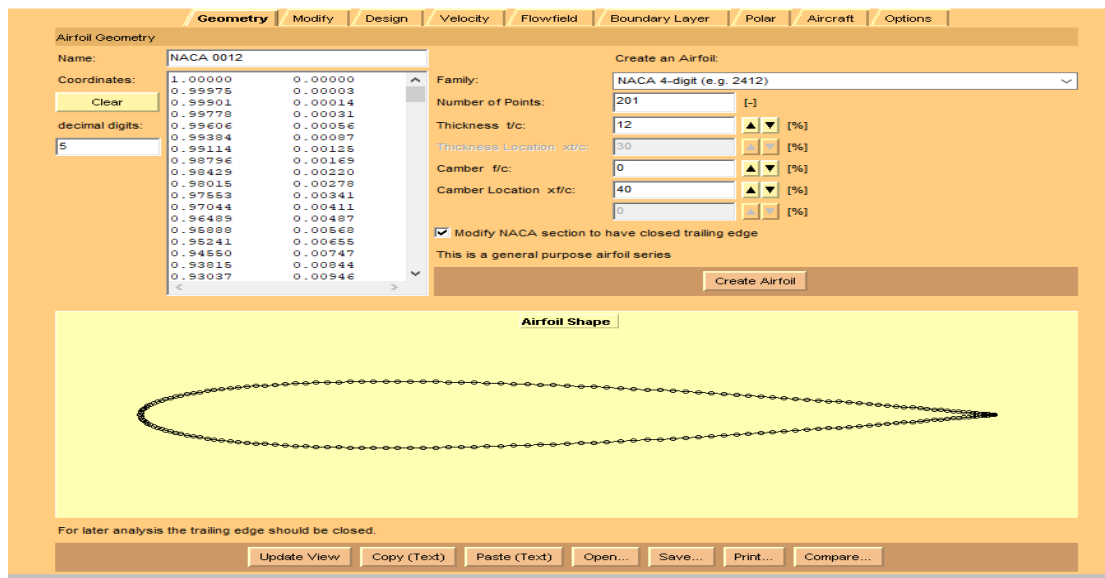


Figure III.2 -Java Foil interface.

III.4 Conclusion

After having all the necessary informations about the software that help us to achieve our goal which is the automation conception of winglet, which will be the object of next chapter.

Chapter IV

GUI REALIZATION

IV GUI realization (Graphical User Interface)

IV.1 Preface

This API is generally stable but some areas are still being added and improved.

The Blender/Python API can do the following

- Edit any data the user interface can (Scenes, Meshes, Particles etc.)
- Modify user preferences, keymaps and themes
- Run tools with own settings
- Create user interface elements such as menus, headers and panels
- Create new tools
- Create interactive tools
- Create new rendering engines that integrate with Blender
- Define new settings in existing Blender data
- Draw in the 3D view using OpenGL commands from Python [23].

IV.2 User interface

Here some tutorials and references list. At the time of this writing the available version of Blender used for this tutorials is the 2.79, let's start from the user interface.

The default layout is composed of individual panels, and inside each one of them we can find an editor. The main editors are

- **Info editor** in the top left part of the screen. It contains the main menu of Blender. It contains also a layout switcher, to quickly change the layout of Blender based on our needs (animation, modeling...) and a renderer switcher, to select the renderer engine.
- **Outliner editor** it contains a list of all the objects in the scene.

- **Properties editor** it contains the properties of an object. It is context specific, so its content will change accordingly to the selected object. It contains also a lot of context specific tabs with specific properties for different context.
- **timeline and animation editor** used to create and modify animation
- **Viewport** that contains the 3D window in which our scene is shown and where we can add, remove or modify objects.

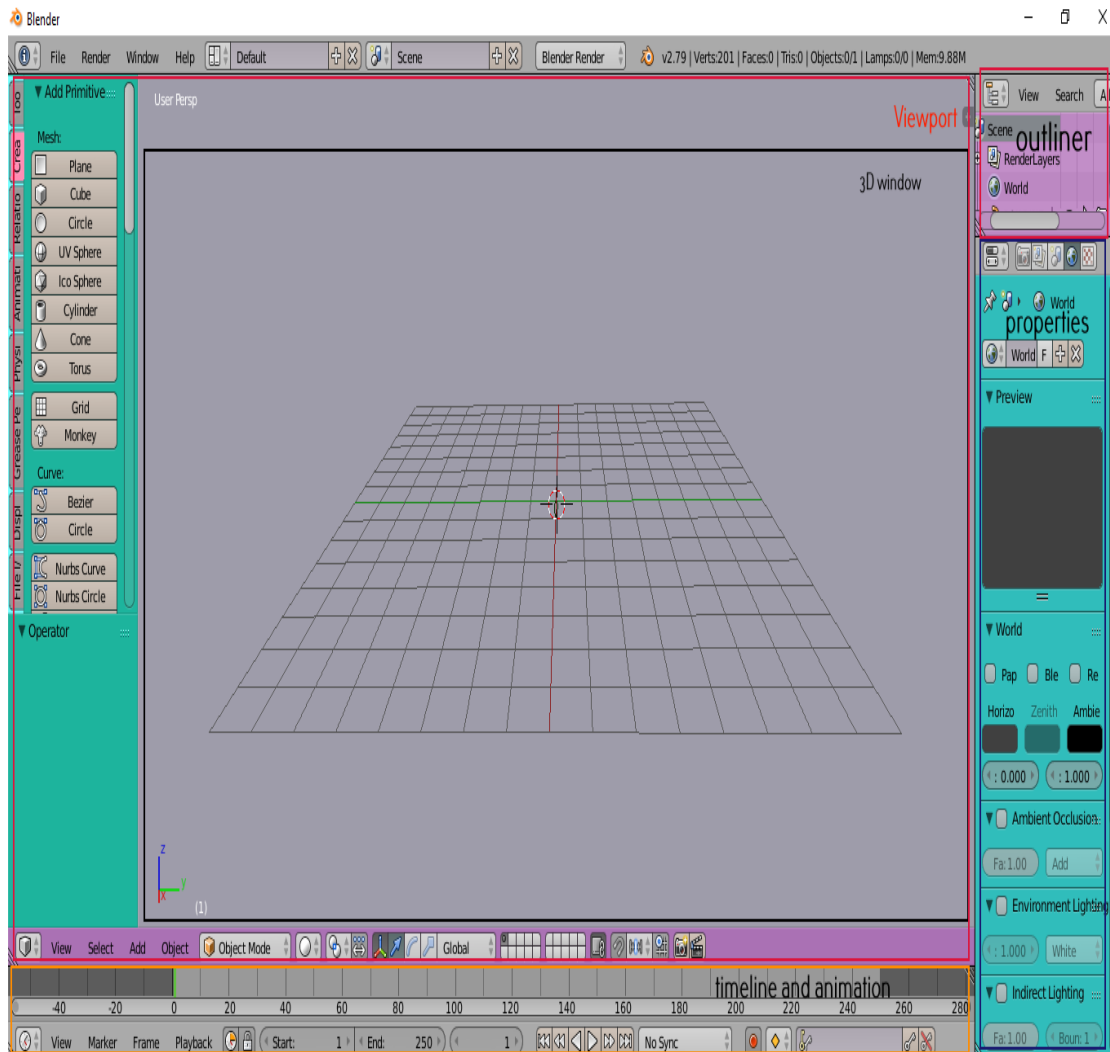


Figure IV.1- Blender editor.

We can switch a panel from one editor to another by clicking on the icon that shows the current editor selected: a list with all the available editors will be shown and we can choose one of them [8].

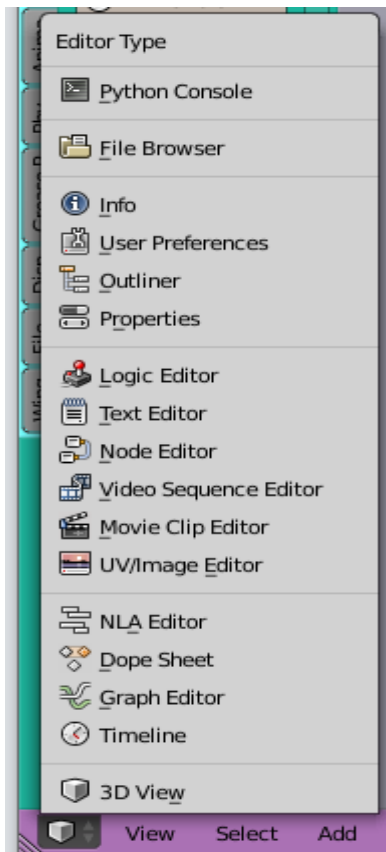


Figure IV.2-The switch editor.

On the left side of the viewport we can find a series of tabs that contain some operations, tools and actions we can apply to the 3D window content. This tabs will change based on the fact that we selected or not an object and also based on which type of object we selected. We can also show the object properties sub-panel by clicking the plus (+) button on the right. That sub-panel gives us some information about the object we selected in the 3D window. At the bottom of the 3D window we can find the 3D manipulator widget, that allows us to scale, rotate and translate object with the mouse drag. Then we have the layer switcher, that will let us create layered scene (we will talk about layers in a future post). We have also the viewport shading button, that let use choose the desired type of visualization we want for our scene :

- **bounding box**
- **wireframe**
- **solid** that shows also colors of the objects.
- **texture mode** that shows also textures of the objects.

- **materials** that let us adds the materials to our objects.
- **rendered**

Finally we have the editing interaction mode selector that allows us to switch between editing mode.

Object mode that allows us to deal with individual objects.

Edit mode that allow us to change the objects.

The menus on the left of the editing mode selector will change accordingly to the mode selected.

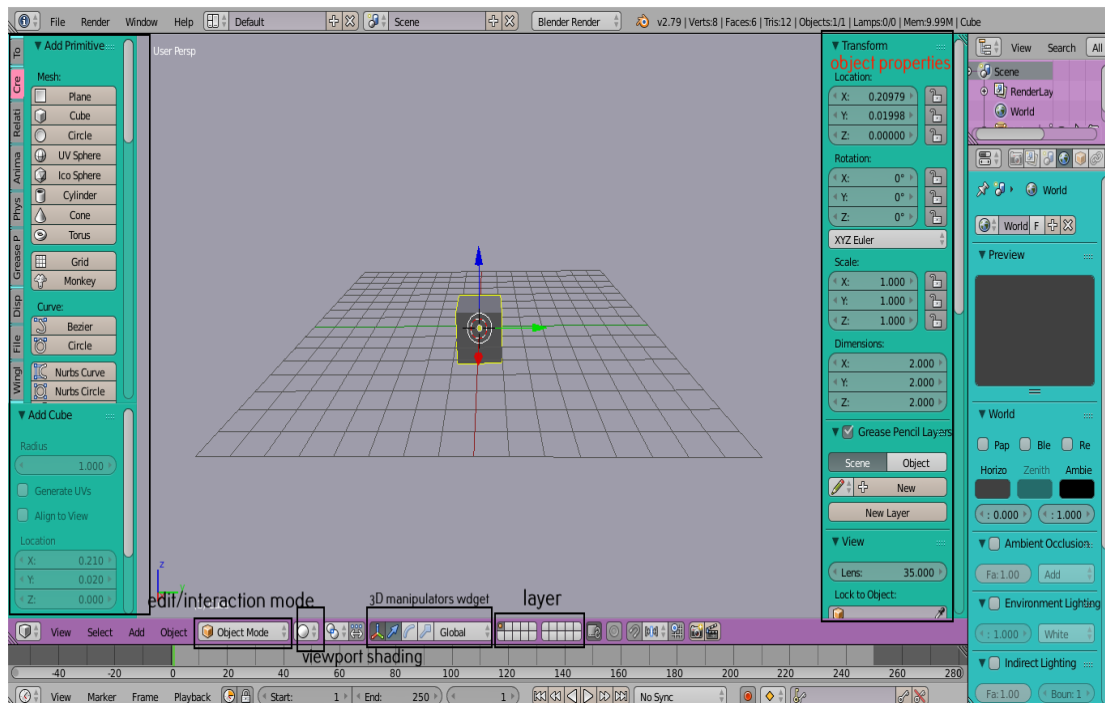


Figure IV.3-The 3D window editor.

To navigate in the 3D space, usually Blender requires a 3 button mouse (we will see below how to emulate a 3 buttons mouse). Anyway, as we're on a MacBook pro we can do the following basic operation with the “alternative” default mapping

- **Orbit around** in the scene by dragging with two fingers
- **Zoom in/out** in the scene with pinch and zoom
- **Pan** in the scene with shift + drag with two fingers

There are also some other basic useful 3D navigation commands:

- Align view to cursor and show all objects with shift + “c” (or alternatively in object mode View -> Align View -> Center Cursor).
- Align view to one side with the options Left, Right, Top, Bottom, Front, Back contained in the View menu.
- Change between orthographic and perspective view with the menu option View -> View Persp/Ortho.

You can change the user preferences by going to File -> User Preferences. Here you can modify settings for:

- **Interface** so what Blender should show in the interface.
- **Editing** so how we edit objects.
- **Input** how mouse and keyboard are configured.
- **Add-ons** where you can manage plugins.
- **Themes** to change the color of the interface.
- **File** to configure standard paths.
- **System** for system specific settings.

To be noted is the option “Emulate 3 Button mouse” in the input settings. This option let Blender emulates a 3 button mouse using the Alt button. In this way you can use this setting to use Blender with standard keys for mouse. That’s enough first post. See you in the second tutorial about selecting and translating objects [8].

IV.3 Winglet and Wing design process

In this project I used blender 2.79 to automate a 3 D wing design and a winglets, this software give a smooth platform to design simple and complicate surfaces and more And javafoil for Airfoil design, 2D simulation of airfoils Design of wing based on Airfoils. One airfoil has different values of lift and drag.

IV.3.1 Blender and Python

IV.3.1.1 Introduction

Python has a simple syntax similar to the English **language**. **Python** has syntax that allows developers to write programs with fewer lines than some other **programming languages**. **Python** runs on an interpreter system, meaning that code can be executed as soon as it is written.

IV.3.1.2 The bpy Module

a. **bpy.ops**

As implied, this submodule contains operators. These are primarily functions for manipulating objects, similarly to the way Blender artists manipulate objects in the default interface. The submodule can also manipulate the 3D Viewport, renderings, text, and much more. For manipulating 3D objects, the two most important classes are `bpy.ops.object` and `bpy.ops.mesh`. The `object` class contains functions for manipulating multiple selected objects at the same time as well as many general utilities. The `mesh` class contains functions for manipulating vertices, edges, and faces of objects one at a time, typically in Edit Mode. There are currently 71 classes in the `bpy.ops` submodule, all fairly well-named and well-organized [19].

b. **bpy.context**

The `bpy.context` submodule is used to access objects and areas of Blender by various status criteria. The primary function of this submodule is to give Python developers a means of accessing the current data that a user is working with. If we create a button that permutes all of the selected objects, we can allow the user to select the objects of his choice, then permute all objects in `bpy.context.select_objects`. We make frequent use of `bpy.context.scene` when building add-ons, as it is a required input to certain Blender objects. We can also use `bpy.context` to access the active objects, toggle between Object Mode and Edit Mode, and accept data from a grease pencil [19].

c. bpy.data

This submodule is used to access Blender's internal data. It can be difficult to interpret documentation on this specific module (the `*/bpy.data.html` page points directly to a separate class), but we will rely heavily on it throughout this text. The `bpy.data.objects` class contains all of the data determining an object's shape and position. When we say the previous submodule `bpy.context` is great for pointing us to groups of objects, we mean that `bpy.context` classes will generate references to data blocks of the `bpy.data` class [19].

d. bpy.types

Contains definitions of all classes that are used in the `bpy.data`, `bpy.context` and `bpy.ops` structures. The blender python API allows integration for:

`bpy.types.Panel`

`bpy.types.Menu`

`bpy.types.Operator`

`bpy.types.PropertyGroup`

`bpy.types.KeyingSet`

`bpy.types.RenderEngine`

This is intentionally limited. Currently, for more advanced features such as mesh modifiers, object types, or shader nodes, C/C++ must be used.

For python integration Blender defines methods which are common to all types. This works by creating a python subclass of a Blender class which contains variables and functions specified by the parent class which are pre-defined to interface with Blender [22].

e. bpy.utils

registration of Blender add-ons, importing Python modules, invoking other programs. Provides utilities for handling the path strings. Contains two additional submodules: `units` and `previews`.

f. bpy.props

functions for creating new class properties, which Blender can display as the controls in the panels (when they are needed). To distinguish them from the ordinary class properties (fields), they are called "Blender custom properties" or just "custom properties" [21].

A. The bmesh Module

In Blender, bmesh objects are fairly heavy-handed and computationally expensive when compared to other core data structures. To maintain efficiency, Blender gives much of the data and instance management work to the user to manage via the API. In general, instantiating a bmesh object requires us to pass a bpy.data.meshes datablock to `bmesh.from_edit_mesh()` while in Edit Mode [19].

B. The OS Module

This module provides a unified interface to a number of operating system functions. Most of the functions in this module are implemented by platform specific modules, such as `posix` and `nt`. The OS module automatically loads the right implementation module when it is first imported.

The built-in `open` function lets you create, open, and modify files [20]

```
sample.au
sample.jpg
sample.wav
...
```

Figure IV.4-Example of Using the OS module to list the files in a directory [20].

C. Function

A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result.

- ✓ In Python a function is defined using the `def` keyword.
- ✓ To call a function, use the function name followed by parenthesis.

- ✓ A parameter is the variable listed inside the parentheses in the function definition.
- ✓ An argument is the value that is sent to the function when it is called.[23]

```
def TheMode(mode_name):
    bpy.ops.object.mode_set(mode=mode_name)
    if mode_name == "EDIT":
        bpy.ops.mesh.select_all(action="DESELECT")
```

Figure IV.5-Example of Function.

D. Booleans

Booleans represent one of two values: **True** or **False**. In programming you often need to know if an expression is **True** or **False**, you can evaluate any expression in Python, and get one of two answers, **True** or **False**. When you compare two values, the expression is evaluated and Python returns the Boolean answer.

- ✓ When you run a condition in an if statement, Python returns **True** or **False**
- ✓ The `bool ()` function allows you to evaluate any value, and give you **True** or **False** in return.
- ✓ You can create functions that returns a Boolean Value.
- ✓ You can execute code based on the Boolean answer of a function.

```
def ProfVis(Etat):
    global ProfList
    Etat = not Etat
    if bpy.context.active_object is None:
        print("No active object")
    else:
        for nm in ProfList:
            print (nm)
            mySelector(nm, False)
            bpy.data.objects[nm].hide = Etat
#=====
def Toggle_Expand(context, State):
    area = next(a for a in context.screen.areas if a.type == 'OUTLINER')
    bpy.ops.outliner.show_hierarchy({'area':area}, 'INVOKE_DEFAULT')
    for i in range(State):
        bpy.ops.outliner.expanded_toggle({'area':area})
        bpy.ops.outliner.item_openclose(all = False)
        #bpy.ops.outliner.show_one_level(open=True)
    area.tag_redraw()
```

Figure IV.6- Booleans function.

IV.3.1.4 Running Scripts

The two most common ways to execute Python scripts are using the built-in text editor or entering commands in the Python console. Both the Text Editor and Python Console are space types you can select from the view header. Rather than manually configuring your spaces for Python development, you may prefer to use the Scripting screen, included default with Blender, accessible from the top headers screen selector. From the text editor you can open .py files or paste then from the clipboard, then test using Run Script The Python Console is typically used for typing in snippets and for testing to get immediate feedback, but can also have entire scripts pasted into it. Scripts can also run from the command line with Blender but to learn Blender/Python this isn't essential [24].

IV.3.1.5 Python Console

The Python Console is a quick way to execute commands, with access to the entire Python API, command history and auto-complete. The command prompt is typical for Python 3.x, the interpreter is loaded and is ready to accept commands at the prompt >>>.

Python Console is a good way to explore the possibilities of Blender built-in Python. The Python Console can be used to test small bits of Python code which can then be pasted into larger scripts [5].

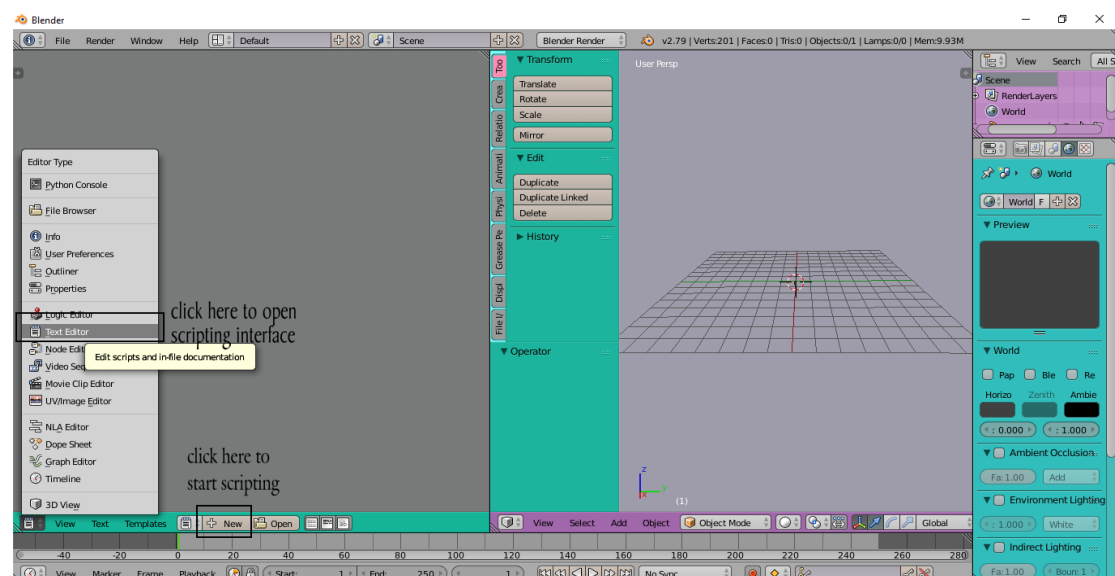


Figure IV.7-Text editor.

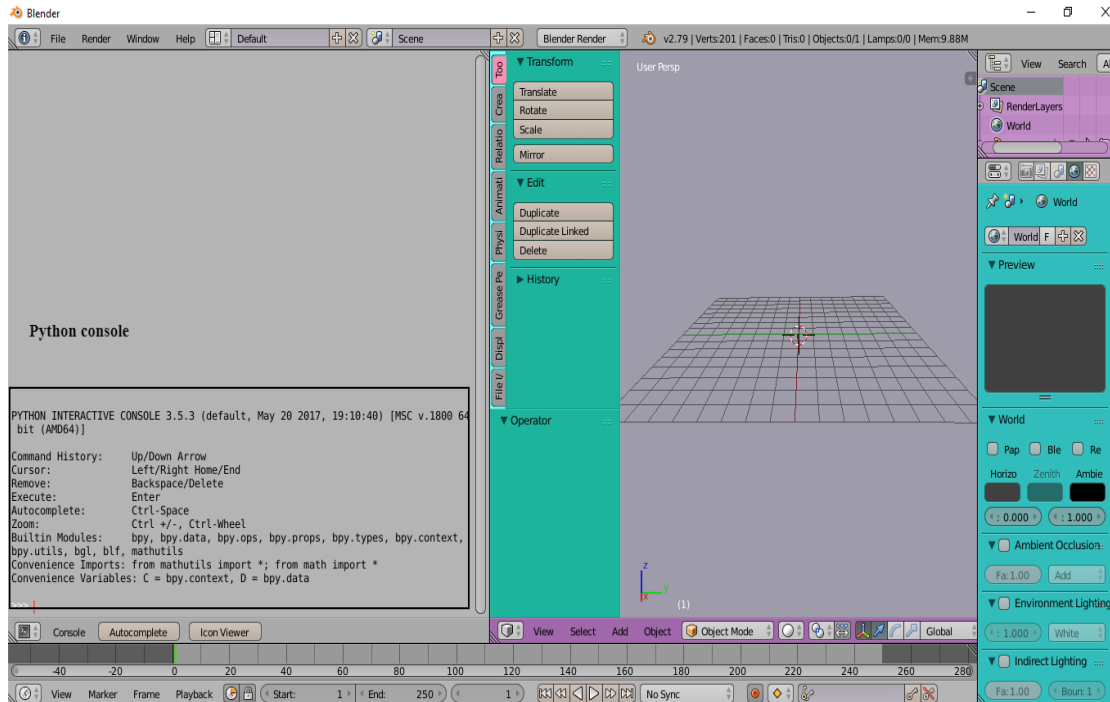


Figure IV.8-Python console.

IV.3.2 Generate the template of an airfoil of desired coordinate points

IV.3.2.1 Methodology

In order to investigate the usage of wingtip devices, the selection of the 2D airfoil to be used for the Construction of the 3D is performed. The 2D airfoil sections for NACA 2412 have been designed using a dxf file by javafoil for importing coordinates to make the profile for the selected airfoils. The 3D wing is constructed by using the dxf coordinate file format in blender 2.79. The values for the wing design have been postulated by closely following Boeing 757-200 's wing. The wing span is measured as 15 m. The wing has also been designed with the wingtip devices; blended winglet, whitcomb, spiroid winglet and semi-circular winglet for the project. The complete wing design is shown in Fig.

IV.3.2.2 JavaFoil

Analysis of Airfoils It is a relatively simple program, which uses several traditional methods for airfoil analysis. Taking a set of airfoil coordinates; it calculates the local, inviscid flow velocity along the surface of the airfoil for any desired angle of attack.

We then used this software to generate the 200 coordinate points of our profile Naca2412 used simultaneously for the wing and the winglet.

1. Download and install Javafoil app.
2. Download the .dat file of the airfoil required [25].
3. Open the javafoil app.

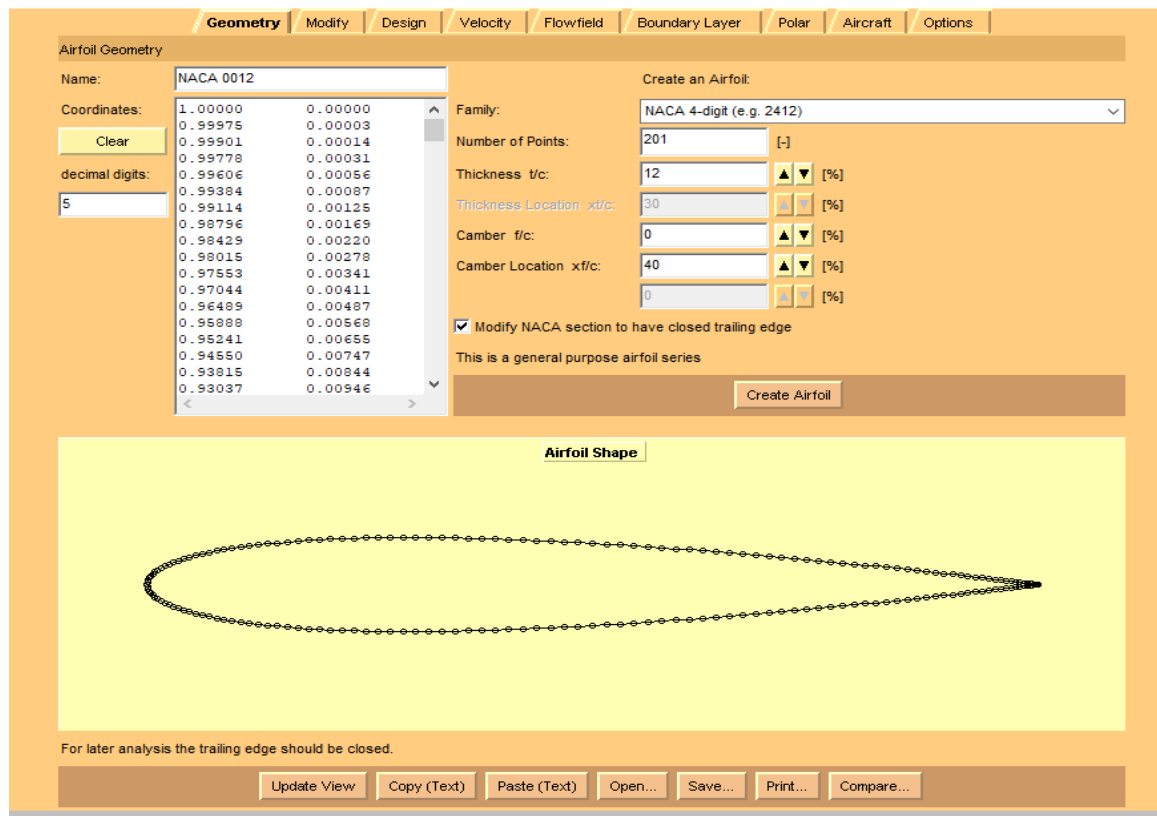


Figure IV.9-Javafoil interface.

4. To open option and open the downloaded dat file to save option and save the airfoil with .dxf extension.

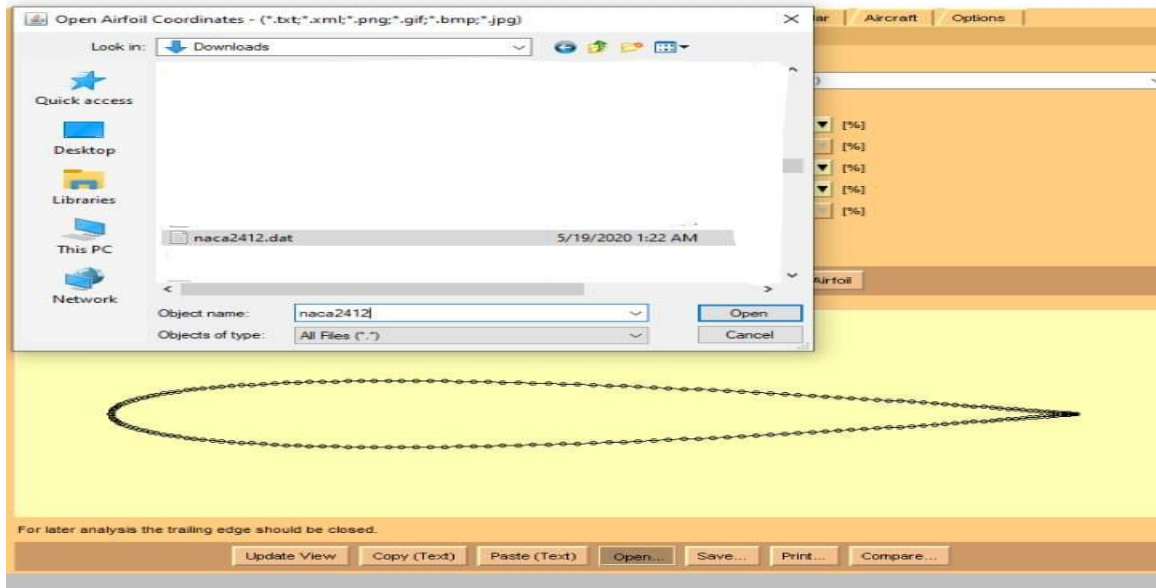


Figure IV.10-Choose location to save the file.

5. Now open the airfoil.dxf in blender.

File → import → AutoCAD DXF

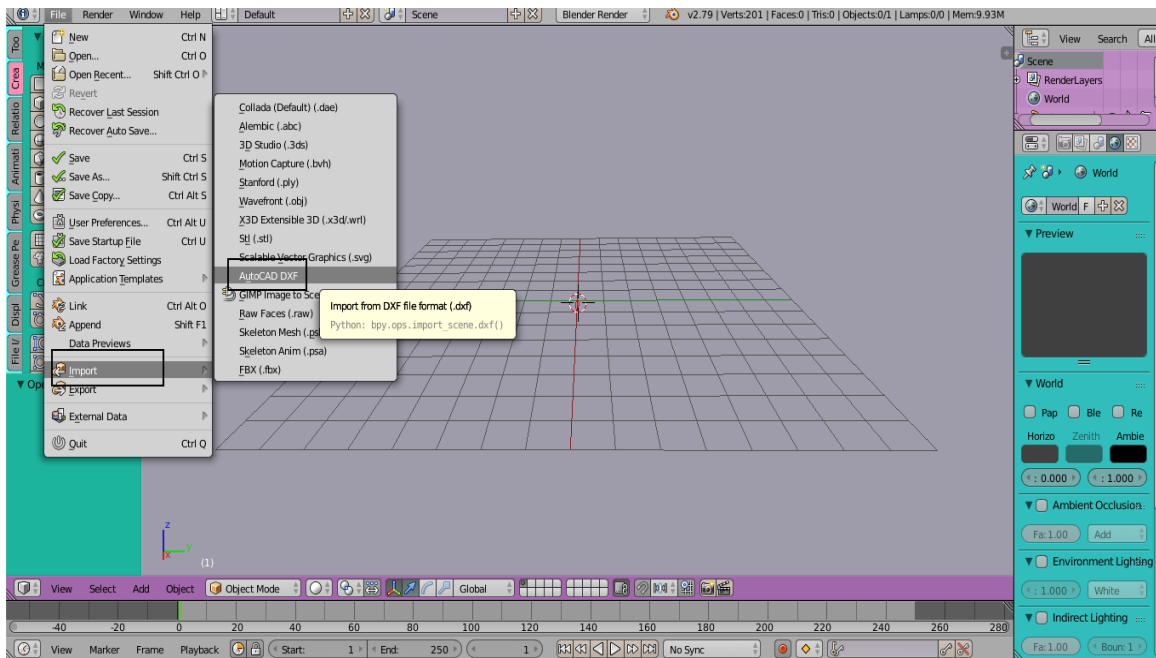


Figure IV.11-Import dxf profile command.

Note: cancel the merged to separate the polyline from the profile and the scale.

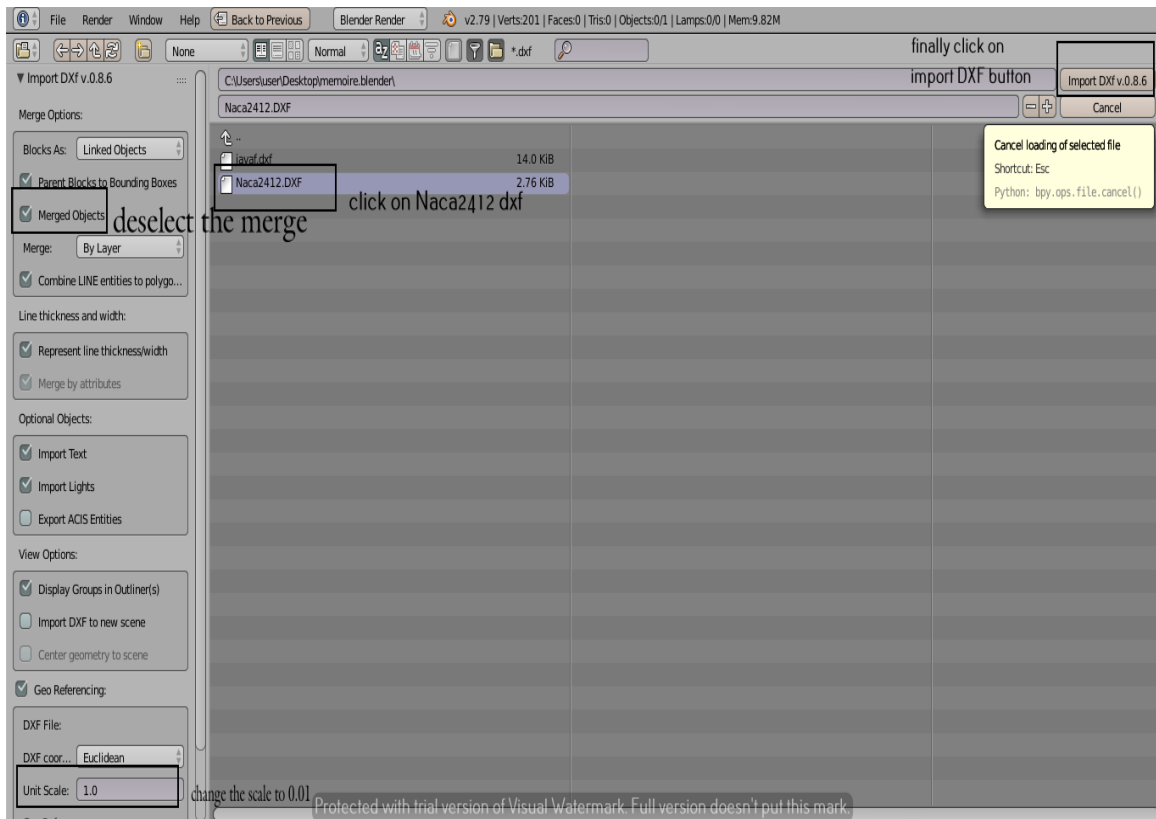


Figure IV.12-Profile dxf importation method.

6. Finally click on **import dxf**

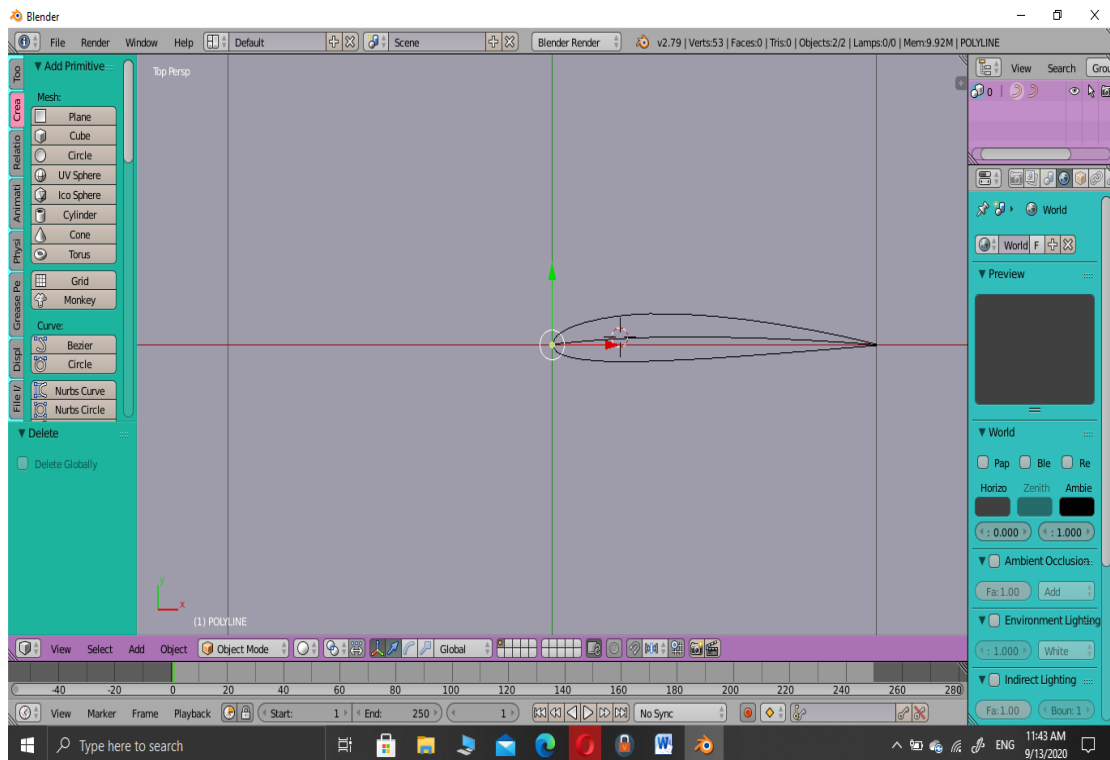


Figure IV.13 -Profile with polyline.

Note: delete the polyline in the profile.

IV.3.3 Running Our Script

- **Modules and Functions Overview**

We begin by giving some background about modules we used.

IV.3.3.1 Importing one Profile

For import profile we used python script to make the work easier, we specified the path and profile name with some modification we have made as shown in the figure below

```
###Import DXF Profile###
class SimpleOperator(bpy.types.Operator):
    """Tooltip"""
    bl_idname = "import_profile.dxf"
    bl_label = "Simple Object Operator"

    def execute(self, context):
        bpy.ops.import_scene.dxf(filepath="F:/Nouveau dossier3/naca2412.dxf", merge=False, dxf_scale="0.01")
        bpy.context.scene.objects['POLYLINE.001'].select=True
        bpy.ops.object.delete()
        bpy.context.scene.objects['POLYLINE'].select=True
```

Figure IV.14-Import profile operator.

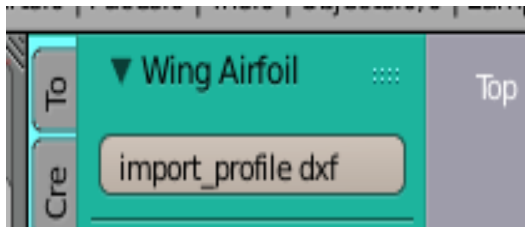


Figure IV.15-Import profile dxf button.

IV.3.3.2 Profiles Realization

Designing a wing need 4 profiles with different cord length and specific distances shown in the figure below

We have entered these parameters to the program to obtain the four profiles we need.

```

27
28 # Initial Profiles Chords
29 Chord_B = 9.0111
30 Chord_R = 7.6042
31 Chord_M = 3.8100
32 Chord_T = 1.3610
33
34 # Initial Span Profiles Positions
35 z_B = 0.0000
36 z_R = 0.7035
37 z_M = 6.2035
38 z_T = 17.7035
39 z_Max = 20.0000
40
41 # Initial Axial Profiles Positions
42 x_B = 0.0000
43 x_R = 1.4069
44 x_M = 5.2011
45 x_T = 7.6501
46 x_Max = 30.0000
47
48 # Initial relative distances (Interesting for design)
49
50 dz_BR = 0.7035 # Span distance between B and R
51 dz_RM = 5.5000 # Span distance between R and M
52 dz_MT = 11.5000 # Span distance between M and T
53
54 dx_BR = 1.4069 # Axial distance between B and R
55 dx_RM = 3.7942 # Axial distance between R and M
56 dx_MT = 2.4490 # Axial distance between M and T
57

```

Figure IV.16-Profiles details.

IV.3.3.3 Creating panels

One of the essential conditions of comfortable work is the convenience and speed of accessing add-ons functional. It is not enough to write add-on and register it in the Blender API system. Be sure to give the easily call add-on functions ability to the user. So, any add-on should determine a place to put interface buttons that provide access to its functionality. Here the class and the operator which is allows making working directory path, filing name and importing Dxf profile buttons.

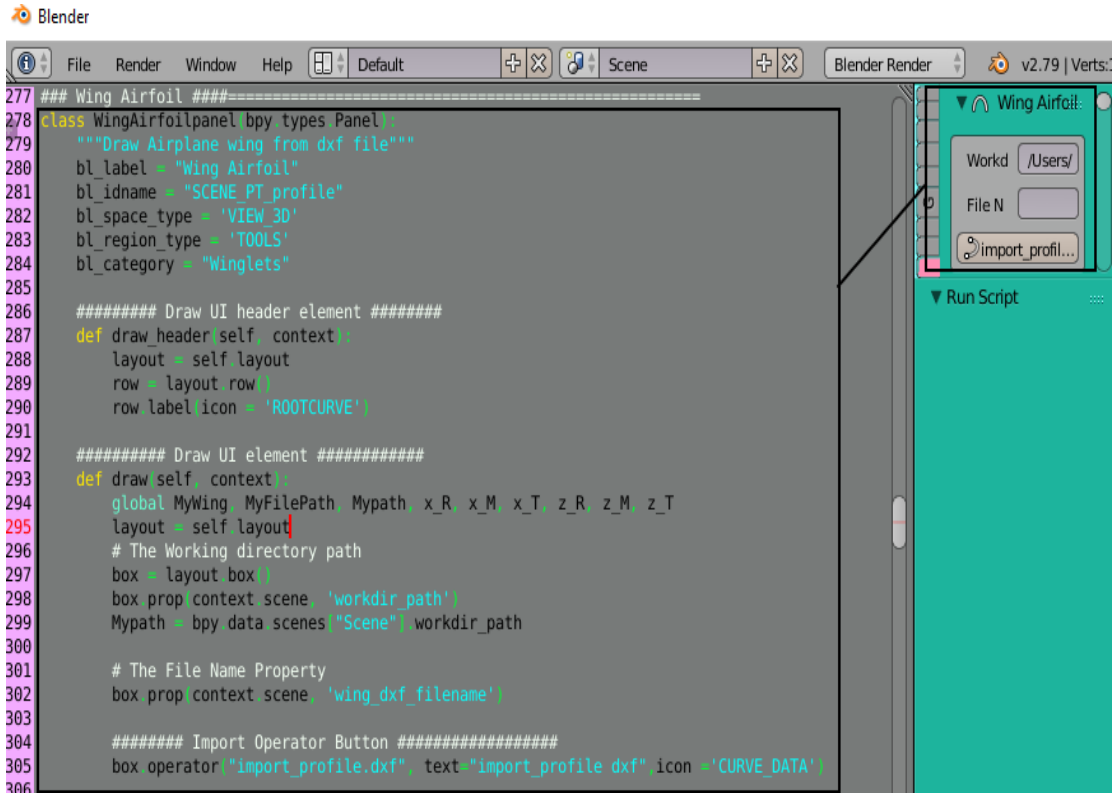


Figure IV.17-Wing airfoil panel.

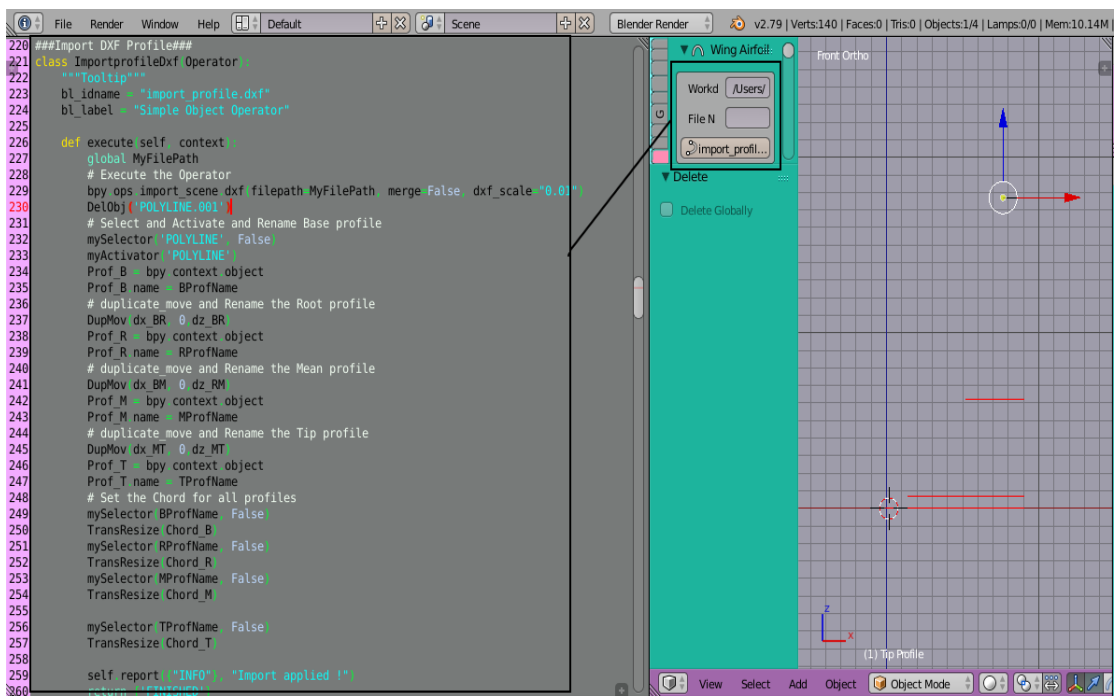


Figure IV.18-Profiles operator.

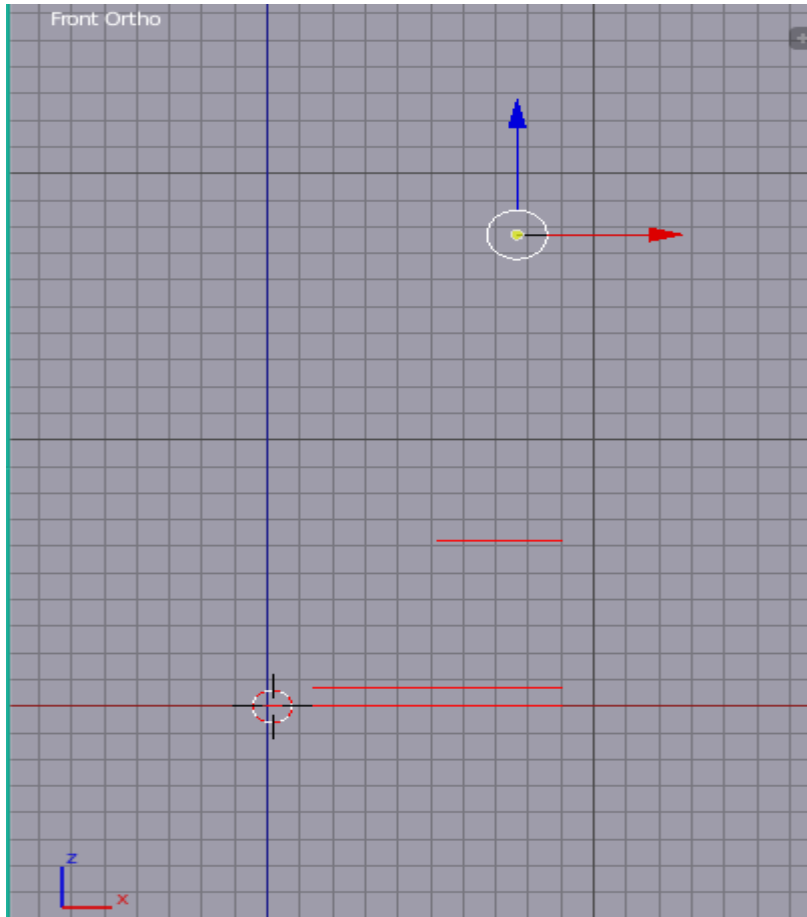


Figure IV.19-The Four Naca2412 profiles.

IV.3.3 Profiles Positions

Other types of wings can be created by moving the location of profiles according to the axis “X” and “Z”.To facilitate the movement very flexible and easy we have added buttons for that.

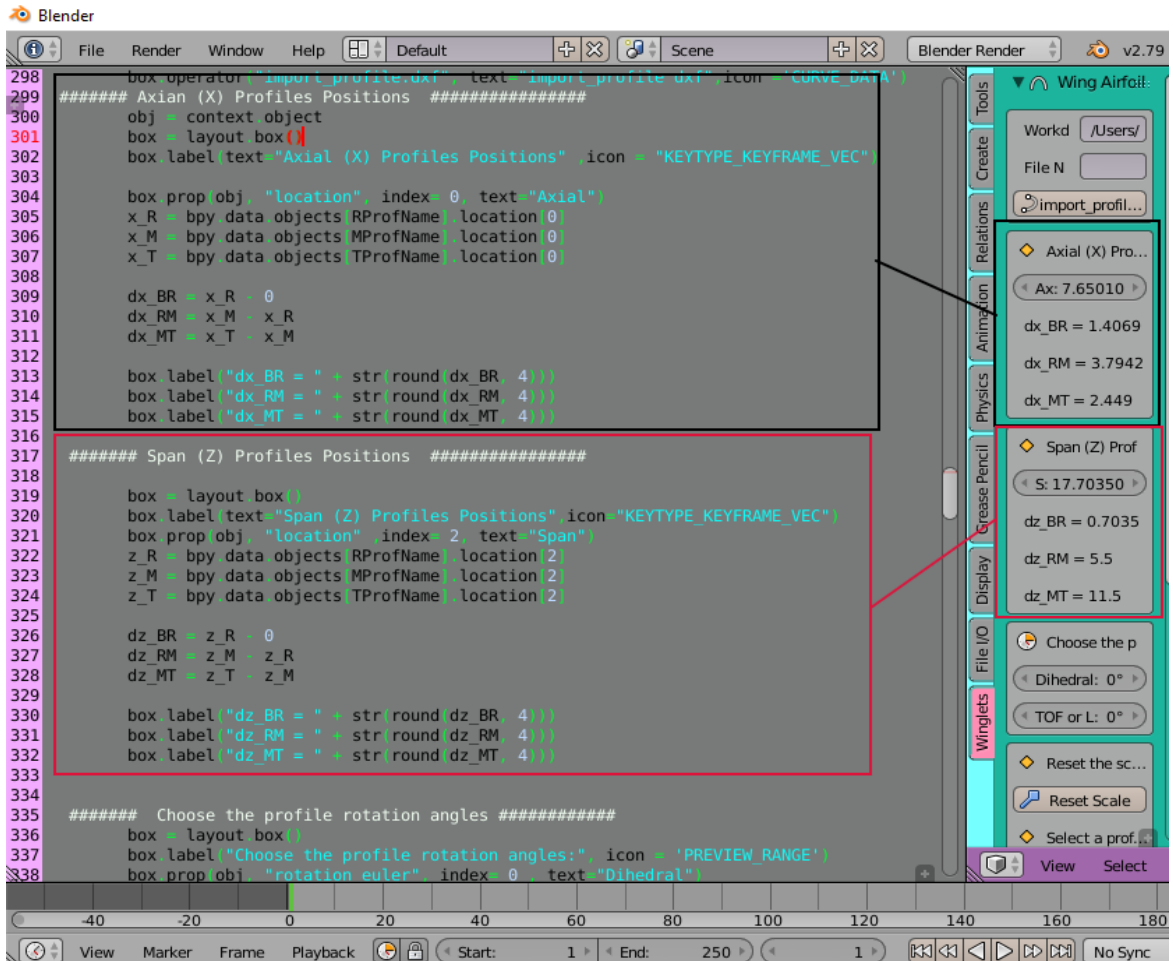


Figure IV.20-Position operator.

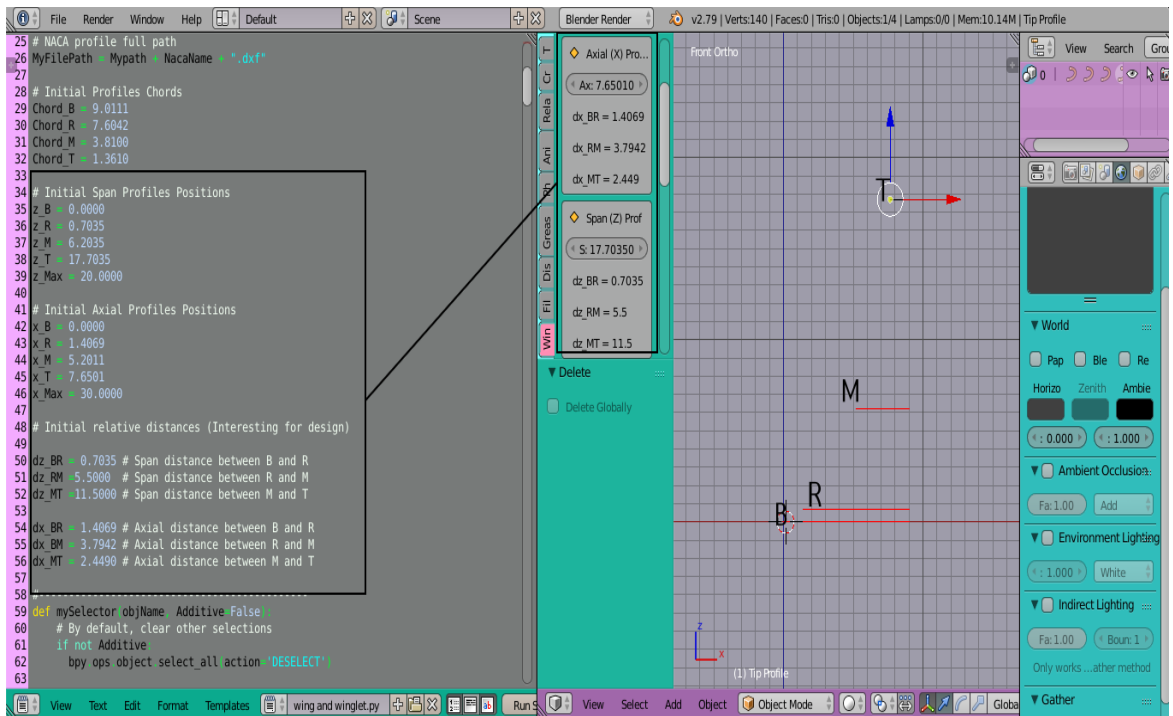


Figure IV.21- Profiles Position Panel.

IV.3.4 Three dimensional drawing of the wing

After the importation of profiles, the process of designing a wing became easy and smooth by adding some specific commands to blender via python codes to get the wing as shown in the figure below.

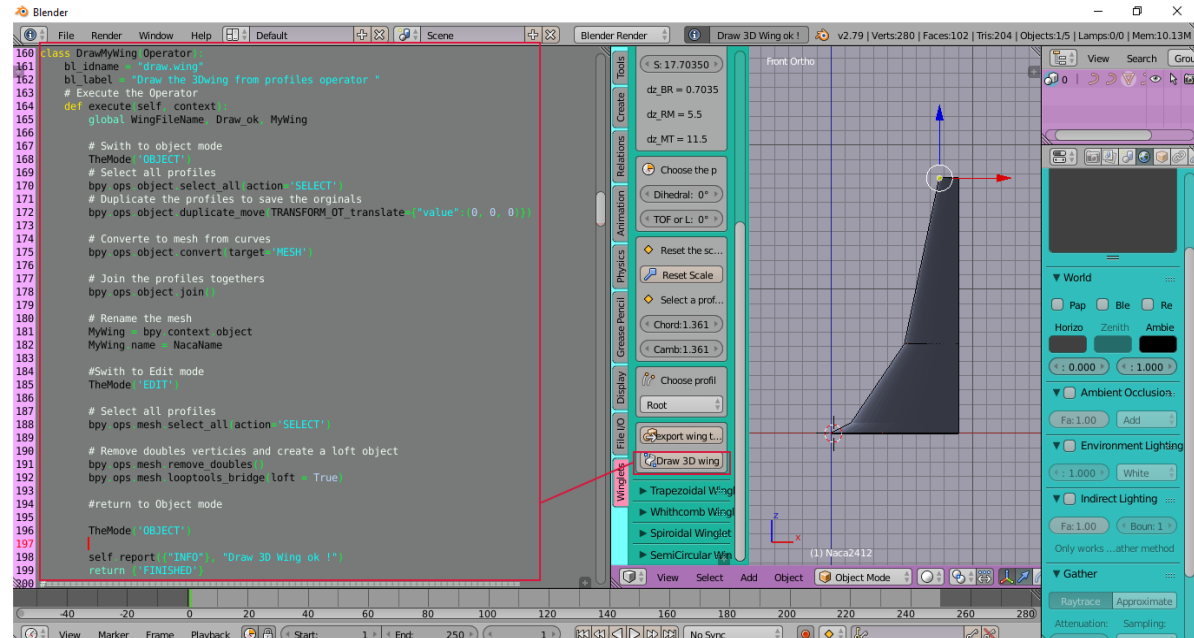


Figure IV.22-Drawing 3D wing operator.

- **Dihedral and take off, Rescale, resize and pick up the profile commands**

Choose the profile rotation angles which mean Dihedral Angle, angle of landing and take off.

Dihedral Angle is the name used by a number of groups whose members provide free air transportation for passengers in need of medical treatment far from home and perform others missions of community service.in aeronautics dihedral are the angle between the left and right wings or tail surfaces of an aircraft Dihedral is also used to describe the effect of sideslip on the rolling of the aircraft



Figure IV.23-Dihedral angle [4].

Planes slowly angle up during **take off** at about 2-3 degrees per second for a Boeing 747. A bit of quick math and using the same Boeing 747 as an example, the average passenger plane has a maximum **take off** angle of about 10-15 degrees. That's well within the plane's tolerances of course.

- ✓ **Reset scale profile** it allows you to **reset** the **size** and orientation of the **profile** to the origin of the scene or to the **profile's** parent
- ✓ **Select a profile and choose a scale** in object mode, select your profile, hit the button of chorde or camber and select scale as you want. The modifiers behave as intended.
- ✓ **Choose profile to variate** you can choose a profile and change its position.

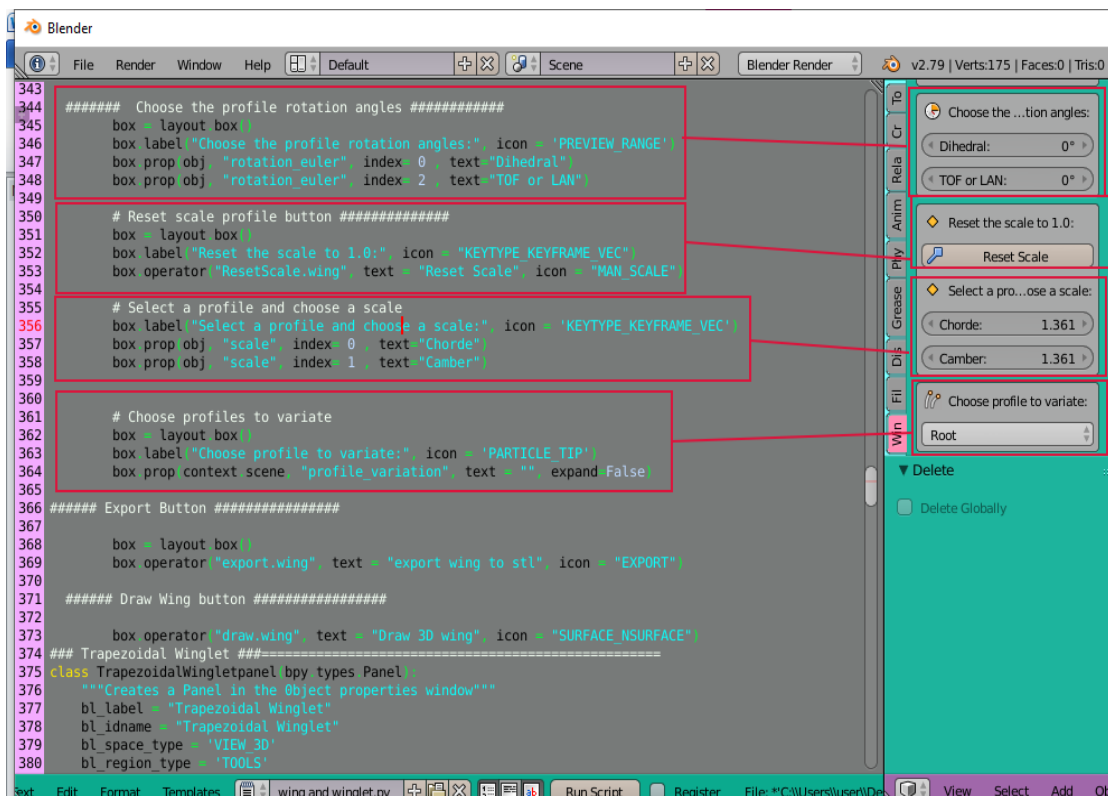


Figure IV.24-Size and position of profiles ui buttons

IV.3.5 Wing Exportation

Blender can export to STL by python code, which is a fairly standard file format for 3D printing. This section demonstrates the changes that occur to a mesh when exporting and re-importing an STL

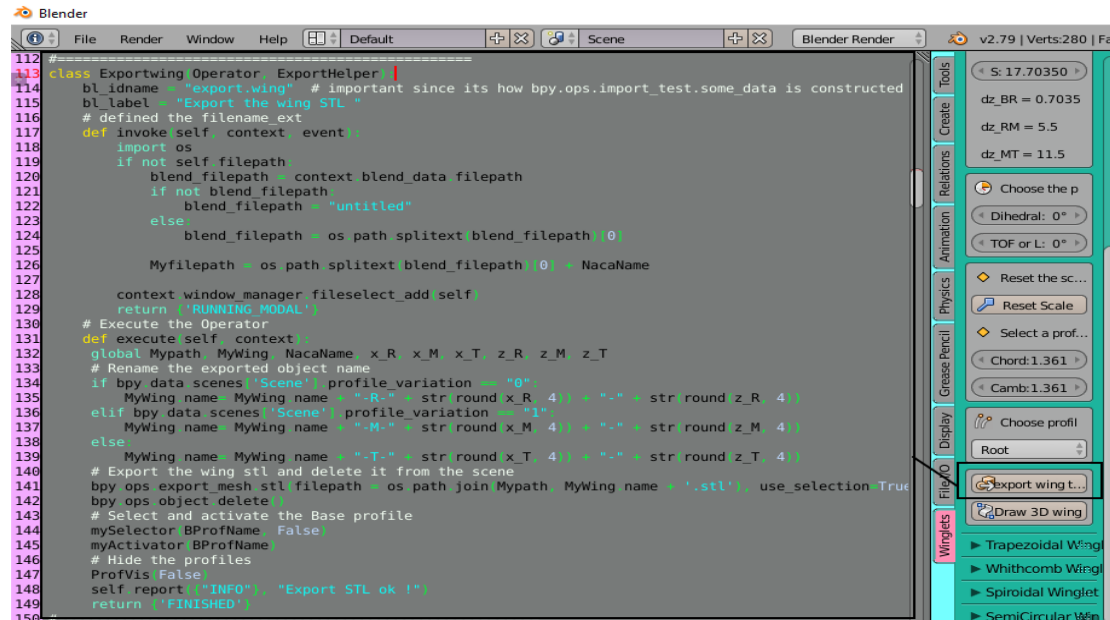


Figure IV.25-Exporting to stl operator.

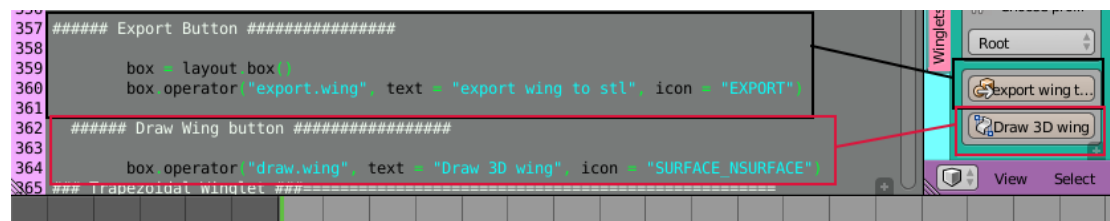


Figure IV.26-Exporting and drawing button.

IV.4 Three dimensional drawing of the Winglet

As we draw the wing previously we followed the steps one by one from profiles to the shape intended we created 3 UI buttons for this winglet, one for import profiles the second one for drawing 3D and finally the third one for exportation, about the design of this type of winglet we mentioned it previously in a short description

```

765 class ImportSemiCircularDxf Operator):
766     """Tooltip"""
767     bl_idname = "import_profile_semicircular.dxf"
768     bl_label = "Winglet Semi_Circular"
769
770     def execute(self, context):
771         global MyFilePath
772         # Execute the Operator
773         bpy.ops.import_scene.dxf(filepath=MyFilePath, merge=False, dxf_scale="0.01")
774         DelObj(['POLYLINE.001'])
775         # Select and Activate and Rename Naca Profile One
776         mySelector('POLYLINE', False)
777         myActivator('POLYLINE')
778         Prof_B = bpy.context.object
779         Prof_B.name = AProfName
780
781         #Select and Activate and Rename Naca Profile Two
782         DupMov(0.6805, 0, 0)
783         Prof_R = bpy.context.object
784         Prof_R.name = CProfName
785         ## Set the Chord for The Naca Profile Name
786         mySelector(AProfName, False)
787         TransResize 0.6805
788
789         mySelector(CProfName, False)
790         TransResize 0.6805
791
792         self.report({'INFO'}, "Import applied !")
793         return {'FINISHED'}
794

```

Figure IV.28-Semi-circular operator.

```

795 ### the Winglet Semi_Circular ###=====
796 class SemiCircularWingletpanel bpy.types.Panel):
797     """Creates a Panel in the Object properties window"""
798     bl_label = "SemiCircular Winglet"
799     bl_idname = "SemiCircular Winglet"
800     bl_space_type = 'VIEW_3D'
801     bl_region_type = 'TOOLS'
802     bl_category = "Winglets"
803     def draw(self, context):
804         global Mypath, Wingletsemi, WingletsemiName, C_y, C_z
805         layout = self.layout
806         scene = context.scene
807         row = layout.row()
808         scene = context.scene
809         box = layout.box()
810         # The File Name Property
811         box.prop(context.scene, 'wing_dxf_filename')
812         ##### Import the Winglet Semi_Circular Button #####
813         box.operator("import_profile_semicircular.dxf", text="import profile dxf", icon='CURVE_DATA')
814         ##### Draw the Winglet Semi_Circular Button #####
815         box = layout.box()
816         box.operator("draw.wingletsemi", text="Draw Winglet Semi-Circular", icon="SURFACE_NSURFACE")
817         obj = context.object
818         box = layout.box()
819         # Change the scale winglet Semi_Circular according to Y
820         box.label(text="Scale Winglet according to Y", icon="KEYTYPE_KEYFRAME_VEC")
821         box.prop(obj, "scale", index=1, text="Y")
822         C_y = bpy.context.object.scale[1]
823         # Change the scale winglet Semi_Circular according to Z
824         box.label(text="Scale Winglet according to Z", icon="KEYTYPE_KEYFRAME_VEC")
825         box.prop(obj, "scale", index=2, text="Z")
826         C_z = bpy.context.object.scale[2]
827         ##### Export Winglet Semi_Circular #####
828         box = layout.box()
829         box.operator("export.wingletsemi", text="Export Winglet Semi-Circular", icon="EXPORT")
830

```

Figure IV.29-Semi-circular panel.

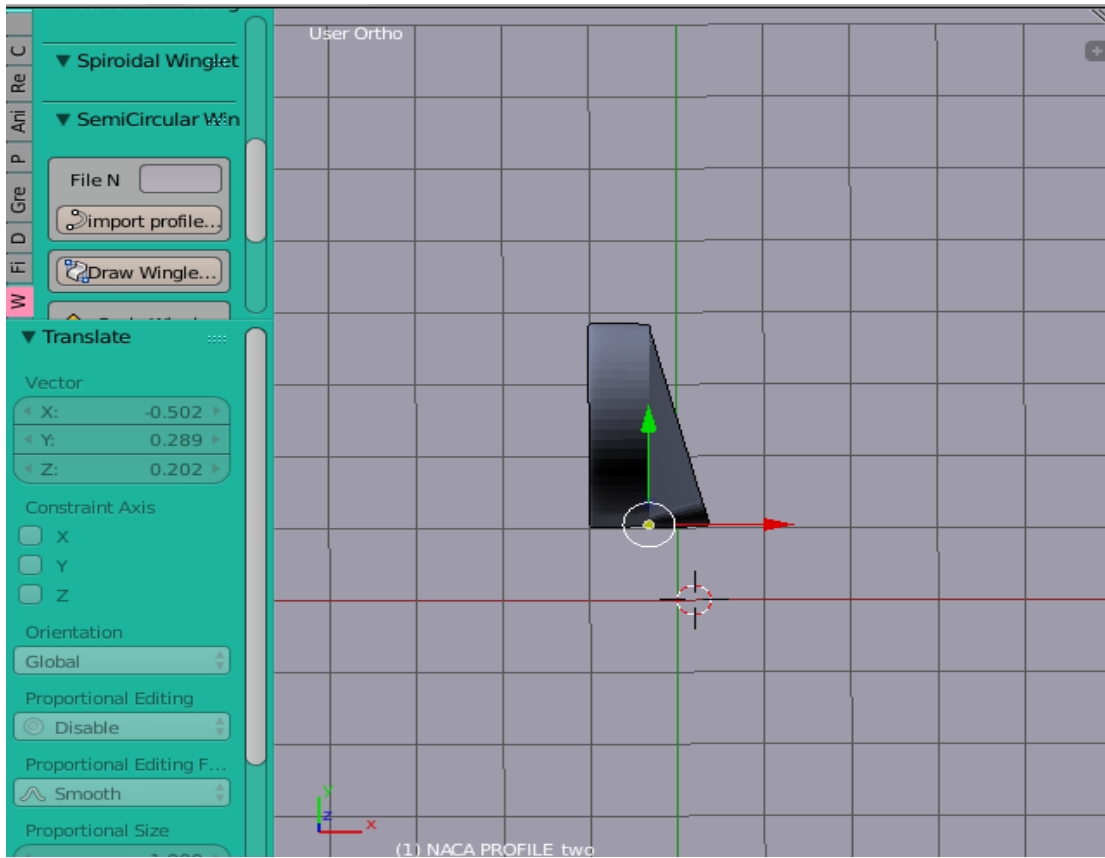


Figure IV.28-Front view of semi-circular winglet.

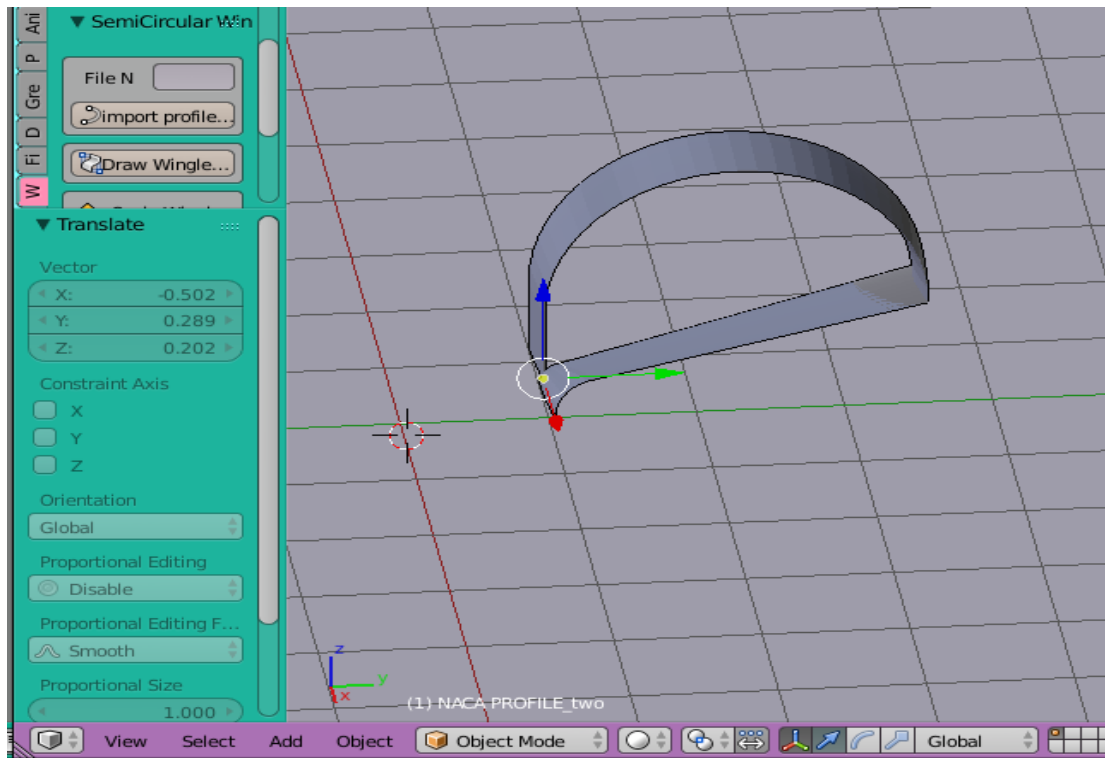


Figure IV.29-Semi -circular left side.

General conclusion

In this study first, we understood the difficulties in the conception of such a complex geometry (Boeing 757-200, and winglets). We have understood and learned the steps to follow to complete both of conception and the graphical user interface. It should be taking in the consideration that we did not easily achieve all these designs and the graphical user interface. The difficulty was using Blender and Python software in March 2020 was the first time we use Blender. by followed some YouTube tutorials about blender and python language to learn some things that helped us start our experience thanks to our supervisor “**Dr laïd MESSAOUDI**” and our efforts with him although in the absence of geometric design data for the wing and the winglets we worked with one of the results that have reached to it in reference [1]. we were able to take the necessary measurements and use "Java Airfoil" software to generate all the points of the NACA 2412 profiles, then we have started by drawing the 757-200 aircraft wing and winglet on Blender, finally, work on the graphical interface, that was successfully building, which make our conception automated The results we found confirm the importance of automated conception, in making the designing process easier for both students and engineers with the ability to modify in the design of the dimensions, through this interface we are ascertaining the best winglet performance and improving the efficiency of the wing. As perspectives to this work, we recommend that the Blended winglet model should be made of a different airfoil and then, drag coefficient reduction could be compared with the old design to choose the efficient winglet. On the other hand, it would be highly interesting to do computational fluids dynamics analysis to those designs that were created from this interface when the aircraft is flying at cruise velocity.

- [1] KENZA LAALAOUI, “Amélioration de l’écoulement autour d’une aile d’avion par l’utilisation de winglet de type trapézoïdale” Department of mechanical Engineering, university BATNA 2, June 22,2016.
- [2] Pritam Jodha, “Aircrafts winglets analysis in CFD”, Independent study and professional development, College of Engineering and Technology, Department of Engineering, University of Derby, 2014-2015.
- [3] Wikipedia
- [4] https://www.studocu.com/in/document/anna-university/aero-engineering_thermodynamics/lecture-notes/m8-selected-pages-basic-aerodynamics/7392868/view
- [5] https://docs.blender.org/manual/en/latest/editors/python_console.html
- [6] Sukhwinder Randhawa, “Open Source Software and Libraries”, January 2008.
- [7] www.innovationpei.com/ebusiness “Open Source Software”, INNOVATION PEI.
- [8] <https://www.fabrizioduroni.it/2018/01/31/blender-tutorial-1-user-interface.html>
- [9] <https://docs.blender.org/api/2.79/>
- [10] A parametric investigation of non-circular spiroid winglets Suhail Mostafaa, Shyam Bose, Archana Nair, Mansoor Abdul Raheem, Thasneem Majeed, Atiqur Mohammed and Young Kim Department of Aeronautical Engineering, Emirates Aviation College, Dubai, UAE
- [11] <https://www.thaitechnics.com/fly/principle.html>
- [12] https://docs.blender.org/manual/en/latest/getting_started/about/introduction.html
- [13] www.innovationpei.com/ebusiness “Open Source Software”, INNOVATION PEI.
- [14] Sukhwinder Randhawa, “Open Source Software and Libraries”, January 2008.
- [15] https://en.wikipedia.org/wiki/Oswald_efficiency_number , Oswald efficiency number, 25 June 2017.
- [16] Flight Test Comparison of Synthetic Vision Display Concepts at Dallas/Fort Worth International Airport
- [17] IOP Conf. Series: Materials Science and Engineering 642 (2019) 012006
IOP Publishing
doi:10.1088/1757-899X/642/1/012006
- [18] <http://staff.univ-batna2.dz/sites/default/files/messaoudi-laid/files/memoire-zekkour.pdf>
- [19] Chris Conlan, Bethesda, Maryland USA, “The Blender Python API: Precision 3D Modeling and Add-on Development”, 2017.
- [20] Microsoft, “Python Standard Library: Core Module”, January 1999.
- [21] Witold Jaworski, “Programming Add-Ons for Blender 2.8” - version 2.0, 2011-2019.
- [22] Jason van Gumster, “Blender For Dummies”, 3rd Edition, 2015.

- [23] https://docs.blender.org/api/2.79/info_quickstart.html
- [24] https://www.w3schools.com/python/python_functions.asp , Python functions.
- [24] https://www.w3schools.com/python/python_booleans.asp , Python Booleans
- [25] https://m-selig.ae.illinois.edu/ads/coord_database.html
- [26] Savanna Rajendran, “Design of Parametric Winglets and Wing tip devices – A Conceptual Design Approach”, Department of Management and Engineering (IEI), Linkoping University Institute of Technology, 2012.
- [27] A CFD Analysis of Wingtip Devices to Improve Lift and Drag Characteristics of Aircraft Wing
Fatima Najeeb Khan¹, Beenish Batul¹ and Ahmad Aizaz¹Published under licence by IOP Publishing Ltd
- [28] <http://www.b757.info/boeing-757-200-specifications/>
- [29] https://www.researchgate.net/publication/24302587_Flight_Test_Comparison_of_Synthetic_Vision_Display_Concepts_at_DallasFort_Worth_International_Airport