DEMOCRATIC AND PEOPLE'S REPUBLIC OF ALGERIA MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH MOSTEPHA BENBOULAID BATNA UNIVERSITY 2

FACULTY OF TECHNOLOGY

Department of Mechanical Engineering

Sector: Aeronautics

# Final Year Project Report

Presented to Obtain the Master's Degree

MAJOR:

AERONAUTICAL PROPULSION

## Automated trapezoidal winglet design

- Name: Saliha BOUNOUAR
- Supervisor: Dr. Laïd MESSAOUDI
- Date: September 2020

# 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 بوينج -200-757 بذون جنيح، ثم الأربع جنيحات (جنيح مائل، نصف دائري، الحلزوني، والوايت كومب) ورسمها في برنامج بلاندر، ثم 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 dessiné 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.
Boeing 757-200 Aircraft wing model without winglet and four different type of winglets (Blended, Whitecomb, 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.

# ACKNOWLEDGEMENTS

*First and foremost, I praise God for leading me on the right path and blessed me with science and knowledge.*

*I would like to thank my supervisor Dr. Laïd MESSAOUDI, my colleagues and friends for their help and support, without whom this project would not have been completed. Also, my mother, father and grandparents rest in peace, my uncle Khilou, my brother Mehdi and my sisters who always supported me in everything.*

# Table of contents

# 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 consummation 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 students 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 consummation, 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 Addon 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 1

# USING WINGLETS

# 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.

## 1.1 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 changes 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.

## 1.2 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 [26].

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. Gratzer 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, Gratzer 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].

## 1.3 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].

## 1.4 Geometric parameters of the airplane wing profile

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 (1.1)

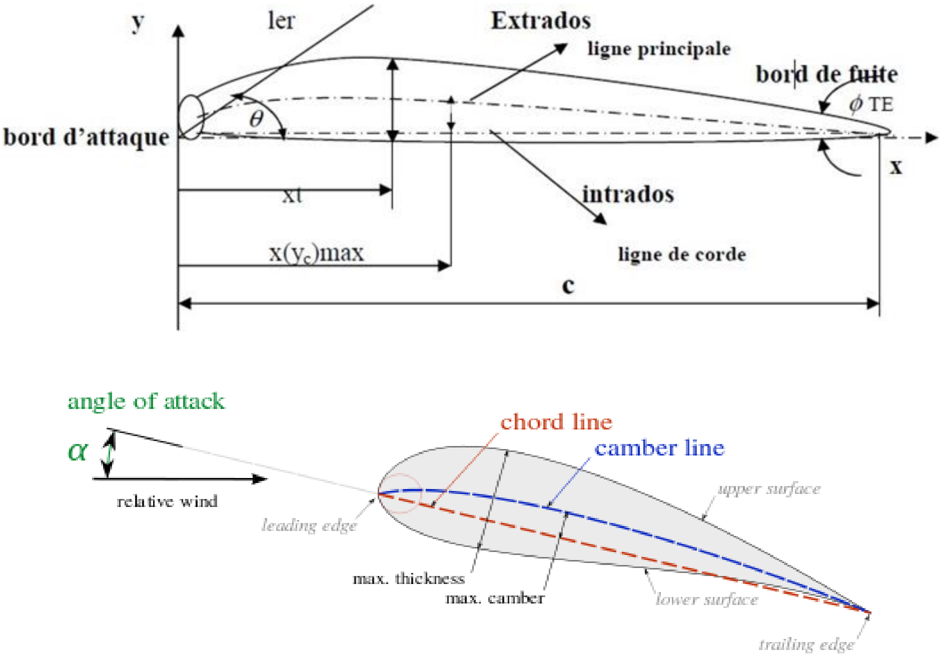


*Figure 1.1- Geometry of the airplane wing profile.*

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 ϕTE (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

ϕTE: 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

$(Y_c)_{max}$: maximum ordinate of the mean line.

$Y_c(x)$: shape of the mean line

$X(Y_c)_{max}$: position of the maximum camber

$\theta$: slope from the leading edge to the trailing edge, equal to the slope between the leading edge and the mean line.

$\Delta Y$ = 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].

### 1.4.1 The NACA airfoil series

The early NACA airfoil series, the 4-digit, 5-digit, and modified 4-/5-digit, were generated using analytical equations that describe the camber (curvature) of the mean-line (geometric centerline) of the airfoil section as well as the section's thickness distribution along the length of the airfoil. Later families, including the 6-Series, are more complicated shapes derived using theoretical rather than geometrical methods. Before the National Advisory Committee for Aeronautics (NACA) developed these series, airfoil design was rather arbitrary with nothing to guide the designer except past experience with known shapes and experimentation with modifications to those shapes [32].

This methodology began to change in the early 1930s with the publishing of a NACA report entitled The Characteristics of 78 Related Airfoil Sections from Tests in the Variable Density Wind Tunnel [32].

In this landmark report, the authors noted that there were many similarities between the airfoils that were most successful, and the two primary variables that affect those shapes are the slope of the airfoil mean camber line and the thickness distribution above and below this line. They then presented a series of equations incorporating these two variables that could be used to generate an entire family of related airfoil shapes. As airfoil design became more sophisticated, this basic approach was modified to include additional variables, but these two basic geometrical values remained at the heart of all NACA airfoil series, as illustrated below [32].

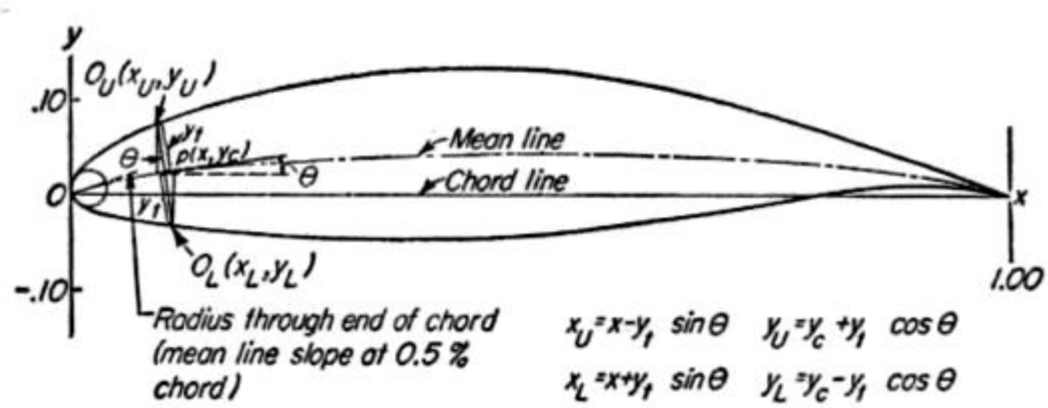

y
.10
0
-.10
O_U(x_U,y_U)
y_t
p(x,y_c)
θ
θ
Mean line
Chord line
y_t
O_L(x_L,y_L)
x
1.00
Radius through end of chord (mean line slope at 0.5 % chord)
x_U=x-y_t sinθ   y_U=y_c+y_t cosθ
x_L=x+y_t sinθ   y_L=y_c-y_t cosθ

*Figure 1.2- NACA airfoil geometrical construction.*

### 1.4.2 Four-digit profile family 4-digit NACA series

In this family, we identify a profile by four digits, the first specifying the maximum relative camber in percentage of the chord, the second indicating the position of this camber in percentage of the chord and the last two figures the maximum relative thickness, always as a percentage of the chord for example, in the case of the NACA 2412 profile:

The 2 represents the maximum relative camber 2

The 4 represents the position of the maximum camber.

The 12 represents the maximum relative thickness [32].

### 1.4.3 Five-digit profile family NACA 5 digits series

In this family, five digits characterize each profile.

1 digit to define the camber by aerodynamic and geometric ways.

2 digits for the thickness.

Law of thickness: same for « 4 digits » Polynomial camber of the 3rd degree + straight Radius of the leading edge: same for « 4 digits »:

r =1.1 $e^2$

For example, in the case of NACA 23015 The 2 represents the characteristic lift coefficient (3/2. 0.2= 0.3).

The 30 represents the position of the maximum camber (1/2 .30 = 15% of the chord) The 15 represents the maximum relative thickness (15%) [32].

Note:

In our project we used the four-digit profile (NACA 2412)

## 1.5 Aerodynamics

### 1.5.1 Aerodynamic forces

Aerodynamic forces are those displayed by the fact of having a body moves at speed immersed in a fluid. There are the following forces in an aircraft:

-Weight, although not strictly an aerodynamic force is always present, and will determine the lift we need to keep the plane in the air. It is a force that is always directed toward the center of the earth. The magnitude of the weight depends on the mass of all the airplane parts, plus the amount of fuel, plus any payload on board (people, baggage, freight, etc.) (Tom Bensom, 2010).

-Lift, the lift is the force perpendicular to the direction of the freestream. To overcome the weight force, airplanes generate an opposing force called lift. The magnitude of the lift depends on several factors including the shape, size, and velocity of aircraft.

-Drag, the drag is defined as the force component in the direction of the freestream. As the airplane moves through the air, there is another aerodynamic force present. The air resists the motion of the aircraft and the resistance force is called drag. Like lift, there are many factors that affect the magnitude of the drag force (Tom Bensom, 2010).

-Thrust, is not an aerodynamic force. To overcome drag airplanes use a propulsion system to generate a force called thrust [3].

The figure reproduced below, shows the forces in an aircraft

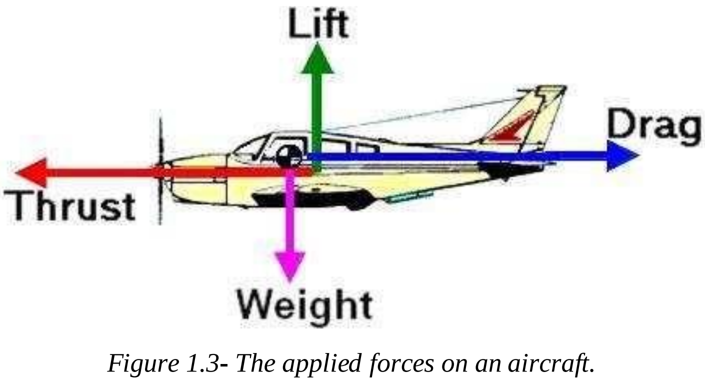


*Figure 1.3- The applied forces on an aircraft.*

The figure reproduced below, shows the drag and lift force in an airfoil.

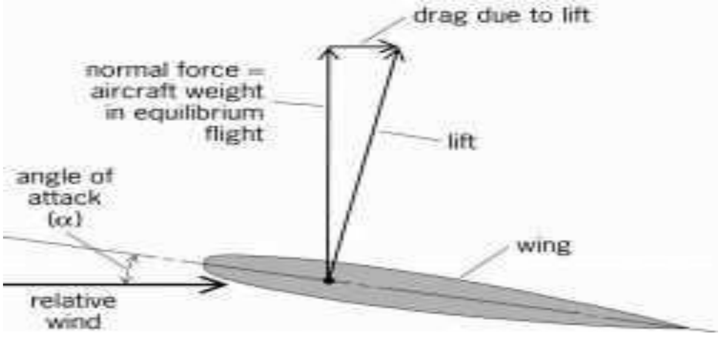


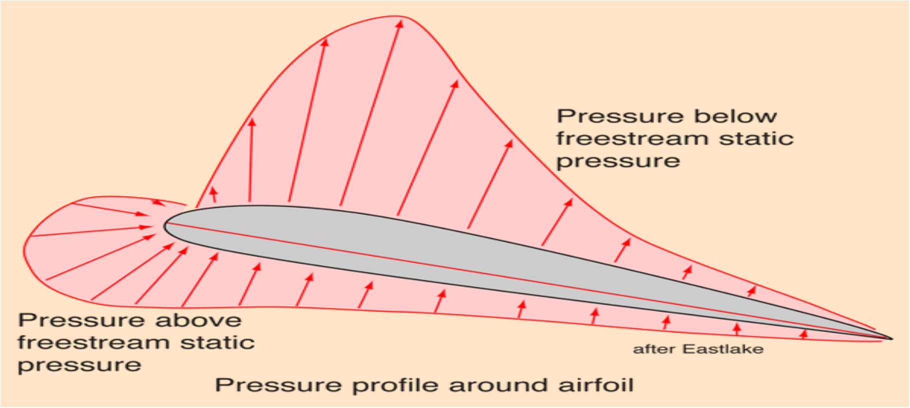
*Figure.1.4- Airfoil lift and drag due to lift.*

## 1.5.2 Lift and drag

When there is a fluid moves through the wing, forcing air particles to narrow at the upper surface of the wing, while the air particles that travel at the lower surface of the wing are expanded. That effect can be seen from Figure (1.3).

Therefore, the flow velocity is increased at the upper surface and the pressure goes down. On the other hand, the flow velocity drops at the lower surface and the pressure grows. The pressure difference between lower and upper surface generates an aerodynamic force called lift.

In the figure (1.4), the pressures caused can be seen in red color


Pressure below freestream static pressure
Pressure above freestream static pressure
after Eastlake
Pressure profile around airfoil

.

*Figure 1.5- Pressure in an airfoil.*

When a stream of air moves over a profile produces a force called aerodynamics. This force is usually decomposed into two components: Lift and Drag. The lift, L, is the component perpendicular to the free flow of air, and L is defined by the following the equation.

$$L = \frac{1}{2}\ r \times v^{2} \times S \times Cl \qquad (1.1)$$

Where:

r: is the density of the fluid through which the body is moving.

V: is the speed of the body relative to the fluid. S is the reference area or wing area.

*Cl:* is the lift coefficient.

With the equation (1.1) is really easy to work out the lift coefficient. The lift coefficient (Tom Bensom, 2009) is a number that aerodynamicists use to model all of the complex dependencies of shape, inclination, and some flow conditions on lift. This equation is simply a rearrangement of the lift equation where we solve for the lift coefficient in terms of the other variables. The lift coefficient *Cl* is equal to the lift L divided by the quantity: density times half the velocity V squared times the wing area S.

The drag force is the resistance force caused by the motion of a body through a fluid, such as water or air. A drag force acts opposite to the direction of the oncoming flow velocity. This is the relative velocity between the body and the fluid.

The drag force *D* exerted on a body travelling through a fluid is given by:

$$D = \frac{1}{2} r \times V^{2} \times S \times Cd \qquad (1.2)$$

Where:

r: is the density of the fluid through which the body is moving.

V: is the speed of the body relative to the fluid.

*S:* is the projected cross-sectional area of the body perpendicular to the flow direction.

*Cd:* is the drag coefficient.

The drag coefficient can be computed from equation (1.2).

The drag coefficient is a measure adimensional that describes the amount of aerodynamic drag caused by fluid flow, used in the drag equation.

Two objects of the same frontal area moving at the same speed through a fluid experience a drag proportional to its *Cd*.

In general, *Cd* is not an absolute constant for a given body shape. It varies with the speed of airflow (or more generally with Reynolds number). A smooth sphere, for example, has a *Cd* that varies from high values for laminar flow to 0.47 for turbulent flow.

A dimensionless number used in aerodynamics to describe the drag of a shape. The drag coefficient, *Cd,* is independent of the size of the object and is usually determined in a wind tunnel.

The lift generates by an airfoil, and by extension, the wings will depend on the altitude and speed aircraft, and the lift coefficient which has our airfoil. There are two different ways to increase the lift:

- increasing the angle of attack or speed.
- Increase the angle of attack causes an increase in the *Cl*, but this has a limit, as speed increases, and that from a certain angle, or velocity, the air starts to come off the wing. When the air begins to travel not attached to the wing, it generates lift.

This effect usually begins at the trailing edge in the airfoils furthest from the fuselage of the plane, and extends to the edge and the center of the plane. To explain why these aerodynamic forces are produced we must use two physical principles: Bernoulli’s principle and continuity equation [3].

## Bernoulli’s principle

$$\frac{p}{\rho} + \frac{1}{2} q^2 + gz = C^{te} \qquad (1.3)$$

## Continuity equation

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} . (\rho \vec{q}) = 0 \qquad (1.4)$$

## 1.6 Drag reduction

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 [4].

## 1.7 Winglet

- ### How it works

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].

- ### 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 Cz 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 Cz and high speed [3].

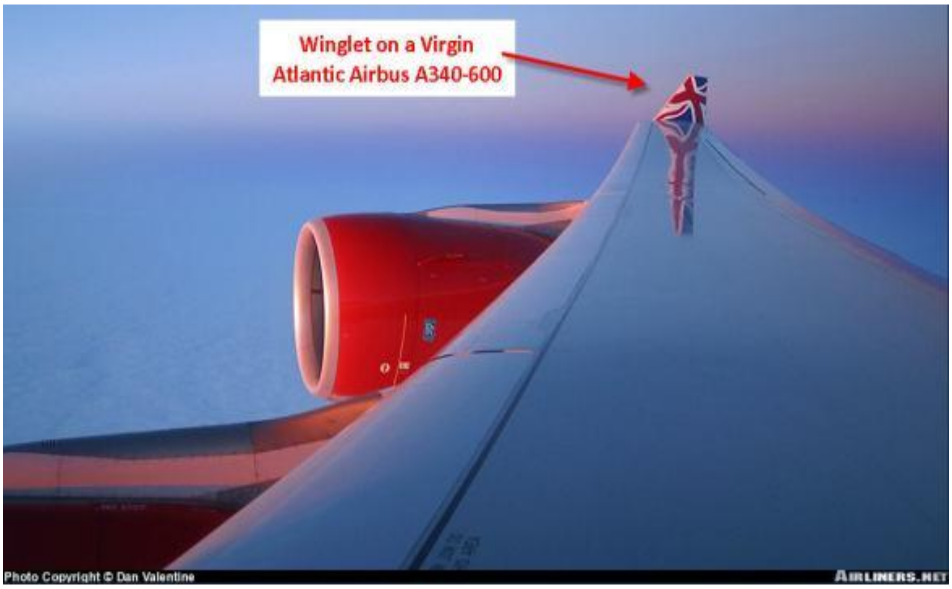

Winglet on a Virgin Atlantic Airbus A340-600
Photo Copyright © Dan Valentine
AIRLINERS.NET

*Figure1.6- Example of winglet.*

## 1.7.1 Measure efficiency of winglet

- 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.

$$C_D = C_{D0} \frac{(C_L)^2}{\pi \, e0 \, AR} \qquad (1.5)$$

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 [6].

## 1.7.2 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:

$$C_D = Cd0 + Cd^2 (C_L)^2 + \frac{(C_L)^2}{\pi\, e\, AR} \qquad (1.6)$$

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 e0 and the lifting-line span efficiency e.

$C_{D0}=Cd_0$

$$\frac{1}{e0} = \frac{1}{e} + \pi\, AR\, Cd^2 \qquad (1.7)$$

For the typical situation $Cd_2>0$, we have **e**0<**e** [6].

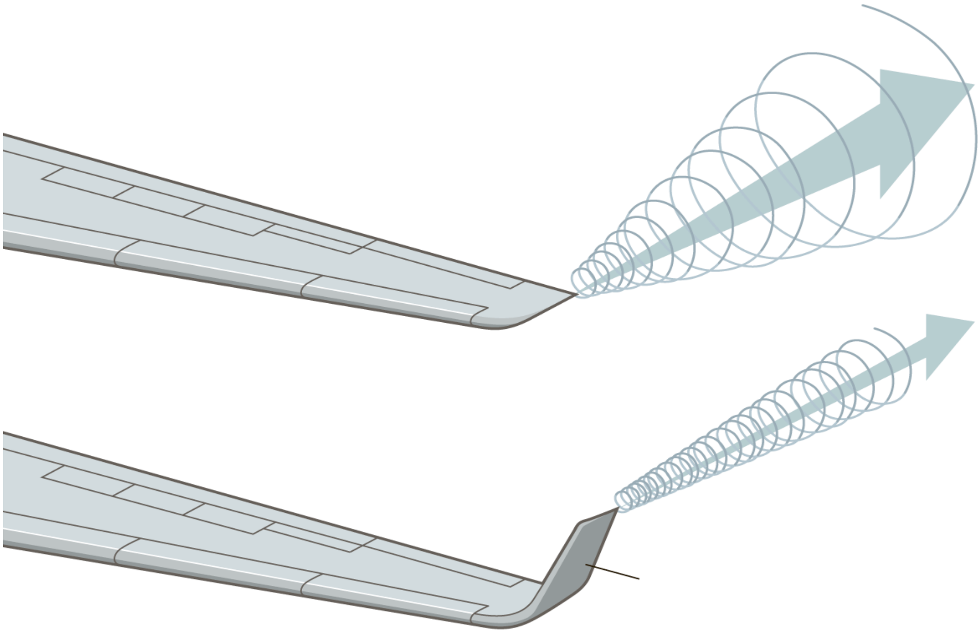


*Figure1.7- Induced drag with and without winglet.*

### 1.7.3 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].

- Wingtip fence

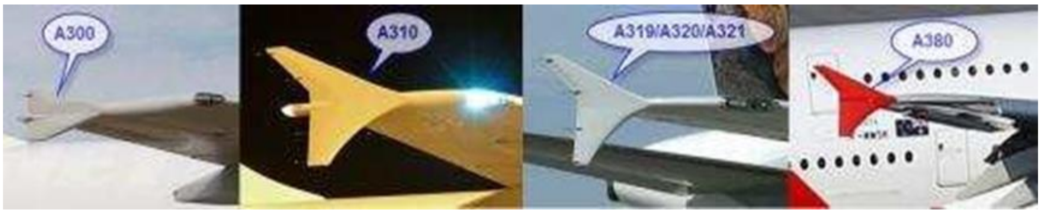


*Figure1.8- Airbus fence Winglets.*

- Blended winglets

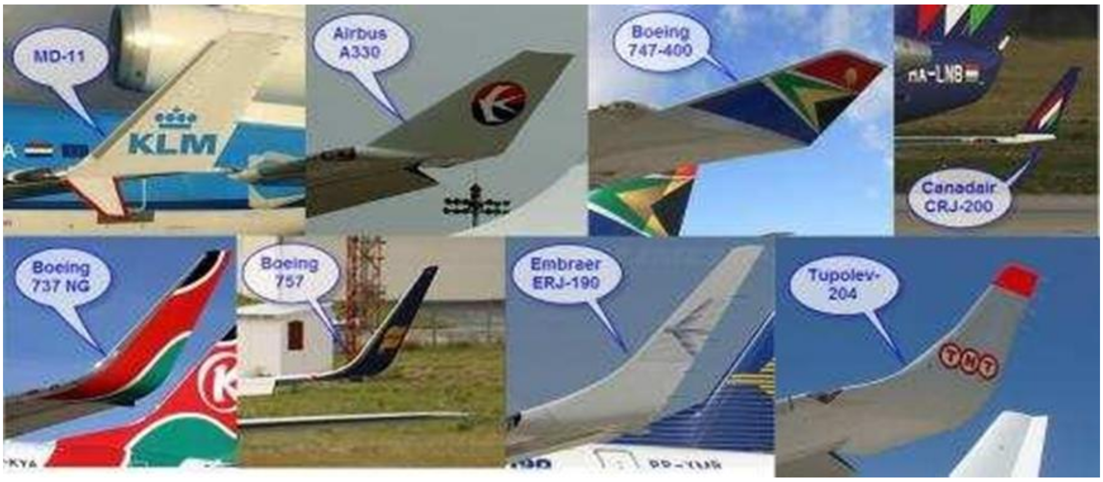


*Figure 1.9- Blended winglets.*

- Raked wingtips

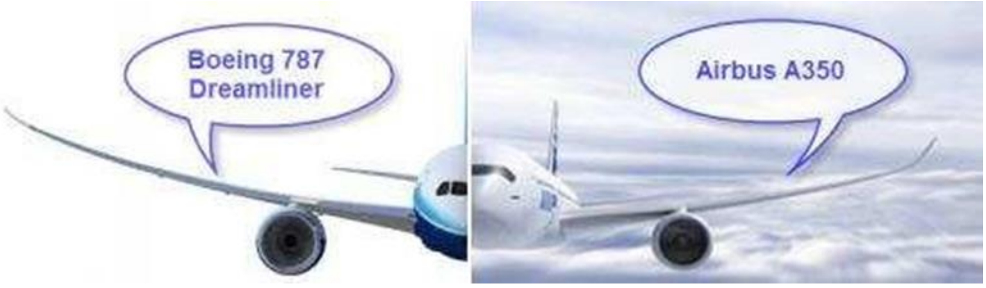


*Figure 1.10- Raked wingtips on the new Boeing 787 and Airbus A350.*

❖ There are other wingtip devices currently in use or in testing (Figure 1.11)

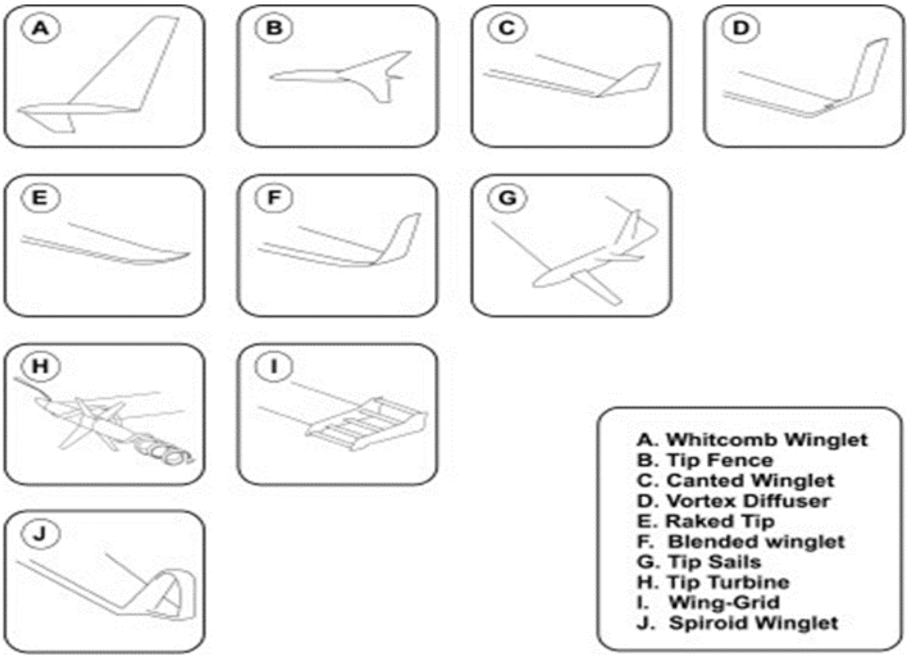


*Figure 1.11- Wingtip devices currently in use or in the testing phase.*

## 1.7.4 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 [4].

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 [4].

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 [4].

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 [4].

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].

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% [4].

Aircraft manufacturers and makers of add-on winglets have also reported improved cruising speeds, time-to-climb rates, and higher operating altitudes [4].

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 [4].

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].

## 1.8 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 2

# BLENDED WINGLET

# Introduction

In this chapter, we will give the necessary information that will allow us to carry out the stages of the winglet design.

## 2.1 Winglet Geometry

### Winglet Aerofoil

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

### Chord Distribution

The small chord distribution demands high CL and when chord distribution is too big then the load of winglet causes outboard section of wing to stall prematurely.

### Winglet Height

The winglet height controls the ideal affected drag and profile drag relationship.

### Twist / Sweep

The angle of Twist/Sweep have similar impact on winglet and they tailor the load distribution. (Shown in figure 2.1 below)

### Toe Angle

The toe angle controls the winglets overall loading. It effects the load distribution on wing and it is ideal for one flight condition.

Toe angle can be seen in figure bellow.

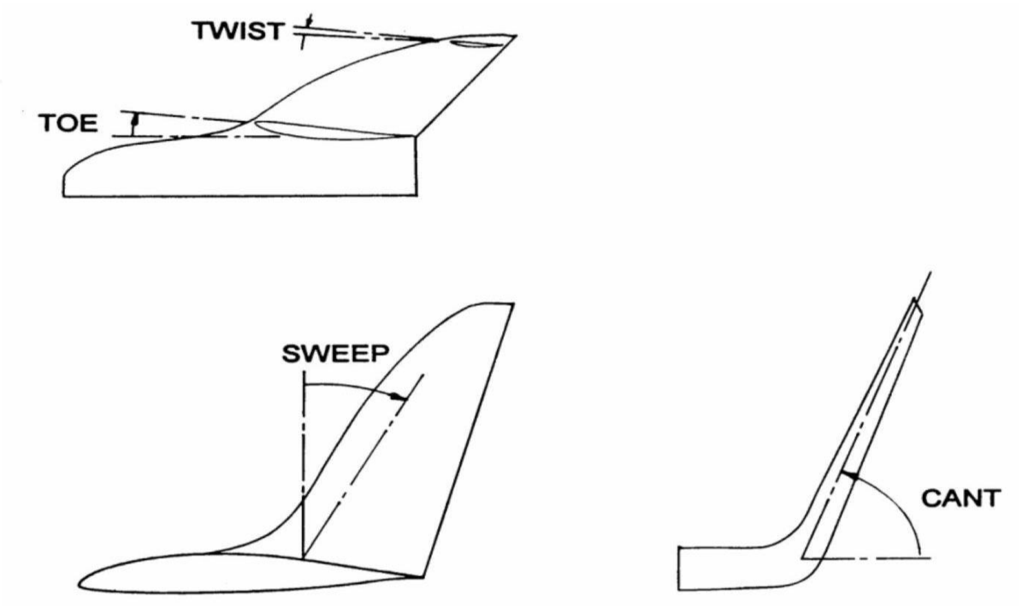


Figure2.1: Image Shows the Geometric quantities used to define a winglet.

## 2.2 Blended winglet

- Description

Blended winglet was developed by Grazter from Seattle in 1994. The unique design in this winglet is no sharp edge found at the wing/winglet intersection and followed by smooth curve. Aviation Partners Inc. (API) and Boeing Company made collaboration in 1999 for the design of advance blended winglets in 1999. Mike Stowell, Executive vice president of APB mentioned about the interference drag, an aerodynamic phenomenon caused due to intersection of lifting surfaces, hence the winglet design was developed to overcome the interference drag formed at the junction of wing and winglet. The winglets were retrofitted in Boeing business jets and also in B737. Now these flights have their services in American airlines (Southwest airlines) and also in European airlines (Ryanair) [26].

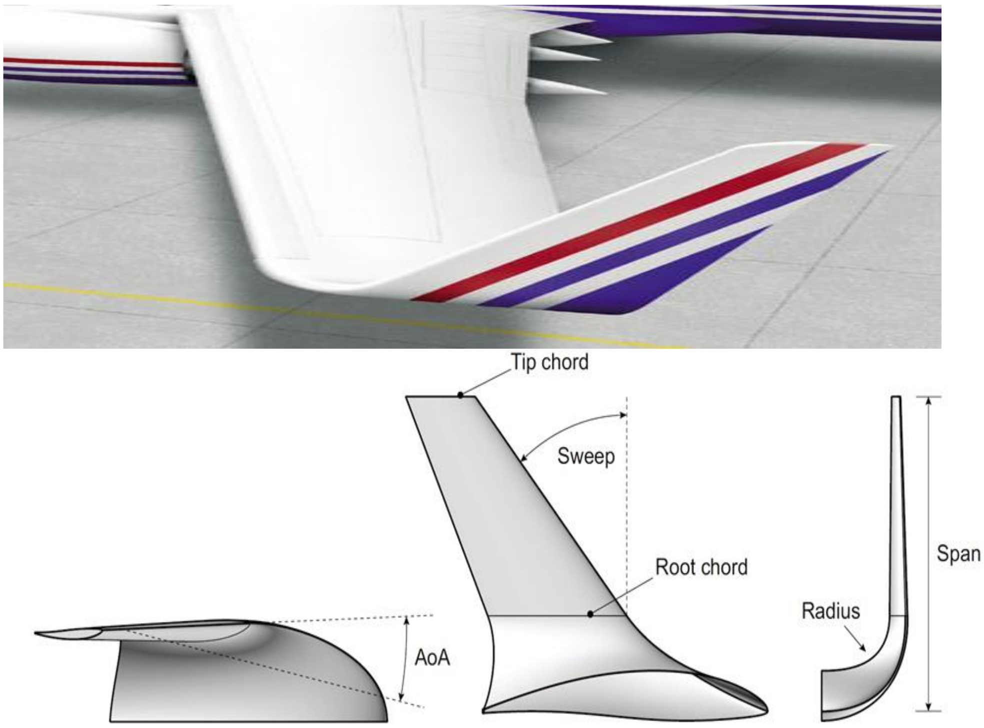


*Figure 2.2- Blended Winglet.*

- Dimensions of Blended winglet

1. 1.83248 m
2. 2.775708 m
3. 0.3525 m
4. 2.198664 m [1].

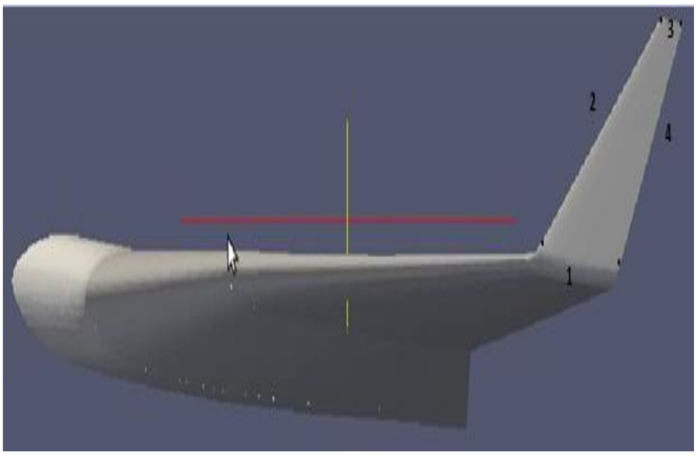


*Figure 2.3- Dimensions of winglet.*

Distance from the wing tip to the start of the winglet:

D: X = 0.28192 m

D: Y = 0.2916 m

D: Z = 0.486 m [1].

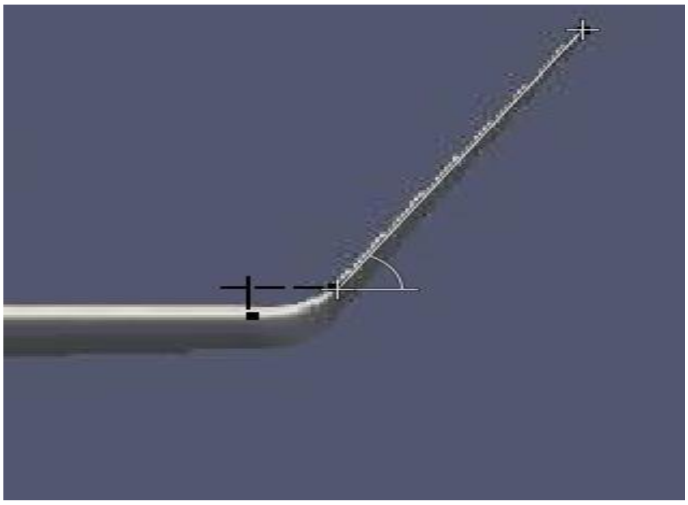


*Figure 2.4- Distance between wing tip and winglet start.*

- The offset this last winglet profile according to Z:

  $\text{COS}\ (60.07) = \frac{x}{2.775708}$

  X= 1.384916 m [1].

## 2.3 Boeing 757-200’s Wing

- Aircraft "Boeing 757-200 " Parameters

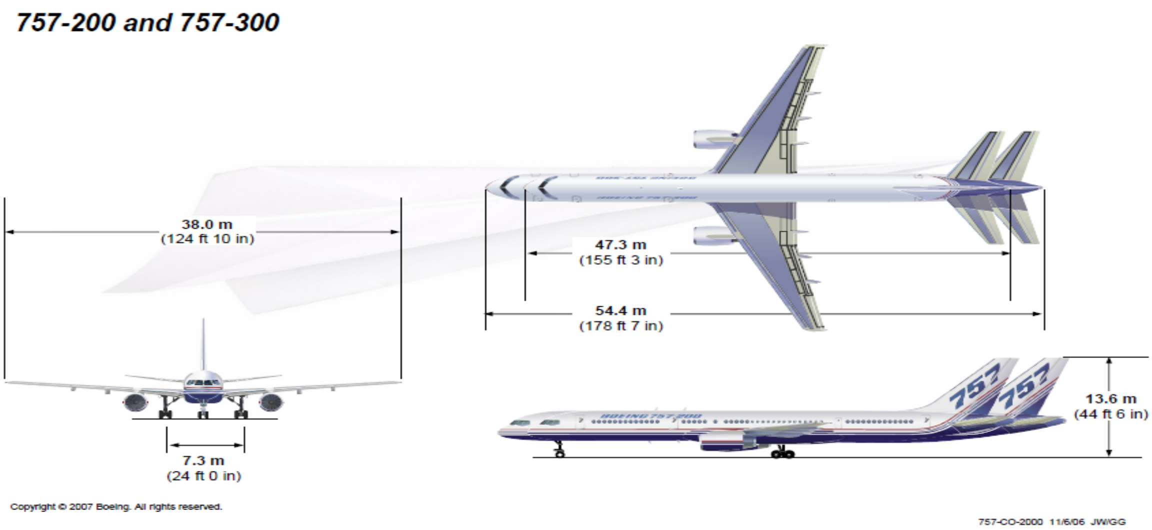


*Figure 2.7- Comparison of dimension between a 757-200 and a 757-300 aircrafts.*

- Wing parameters

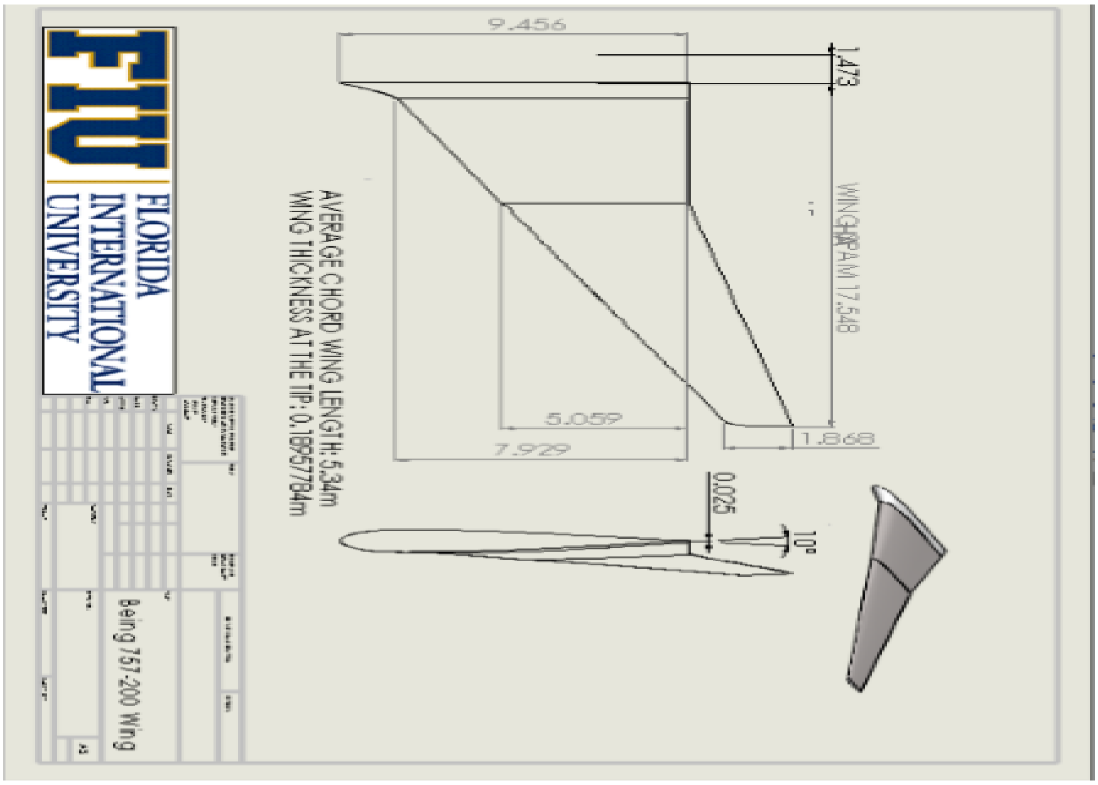


*Figure 2.8- Technical drawings of the wing for a 757 Boeing aircraft.*

## 2.4 Conclusion

After collecting the necessary dimensions that we will use them to concept our wing and winglet, we will have some background about the software and tools in the next chapter.

# Chapter 3

# PROGRAMING TOOLS

# 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 an open source software, therefor in this chapter we will provide you the necessary information about the software we used on the modeling and on creation of our common winglet interface.

## 3.1 Open source software

### 3.1.1 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 [8].

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 [8].

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 [8].

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 [8].

### 3.1.2 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 [8].

### 3.1.3 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 [8].

### 3.1.4 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 [7].

- 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 [7].

- 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) [7].
- 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 [7].

- 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 [8].

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.

## 3.2 Blender

### ❖ 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].

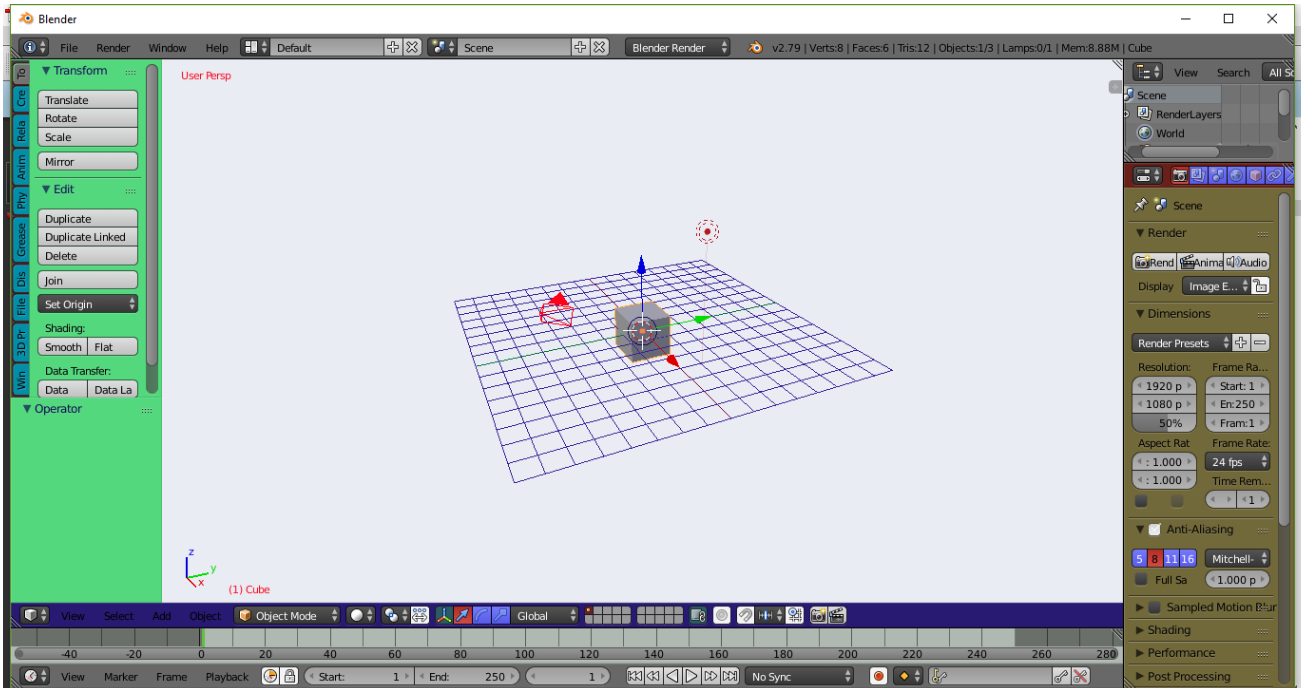


*Figure 3.1- Blender Interface & 3D View Header.*

## 3.3 Python

### ❖ What is Python? Executive Summary

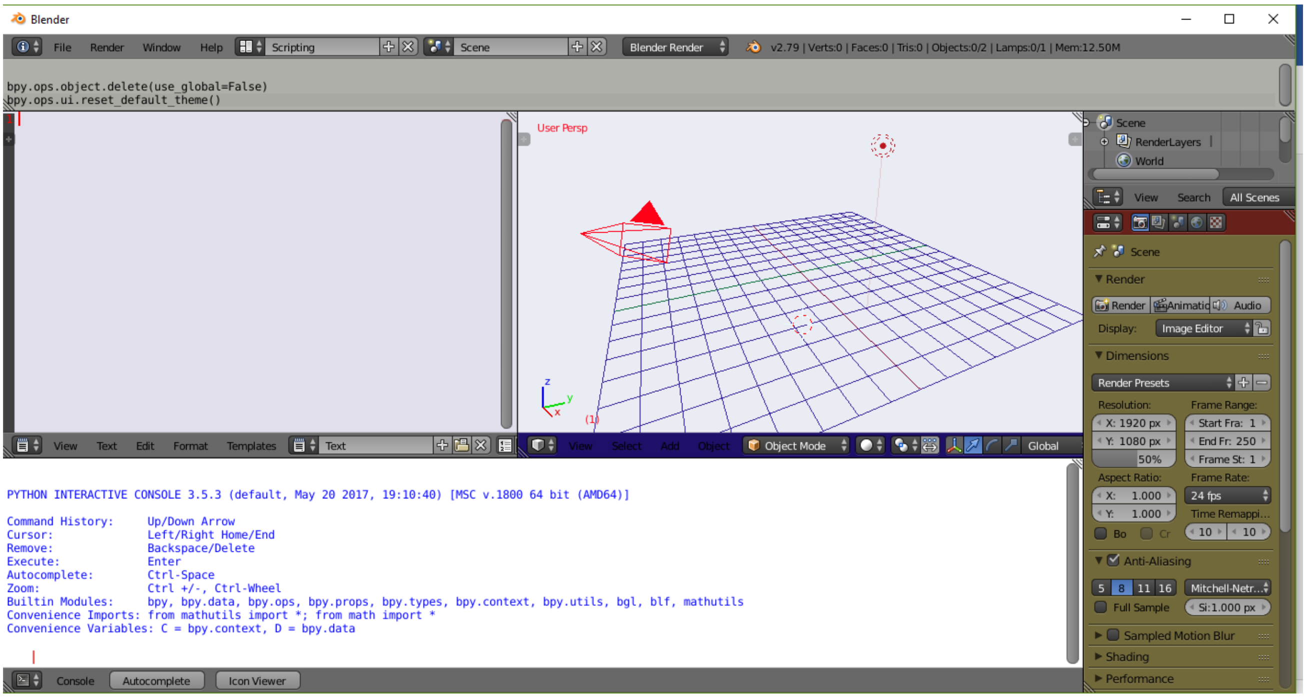
The Python programming language was created in 1989 by Guidovan Rossum, in the Netherlands. The name Python comes from a tribute to the television series Monty Python's Flying Circus of which G. van Rossum is a fan. The first public version of this language was published in 1991.

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 [13].

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].



```
bpy.ops.object.delete(use_global=False)
bpy.ops.ui.reset_default_theme()
```

```
PYTHON INTERACTIVE CONSOLE 3.5.3 (default, May 20 2017, 19:10:40) [MSC v.1800 64 bit (AMD64)]

Command History:     Up/Down Arrow
Cursor:              Left/Right Home/End
Remove:              Backspace/Delete
Execute:             Enter
Autocomplete:        Ctrl-Space
Zoom:                Ctrl +/-, Ctrl-Wheel
Builtin Modules:     bpy, bpy.data, bpy.ops, bpy.props, bpy.types, bpy.context, bpy.utils, bgl, blf, mathutils
Convenience Imports: from mathutils import *; from math import *
Convenience Variables: C = bpy.context, D = bpy.data
```

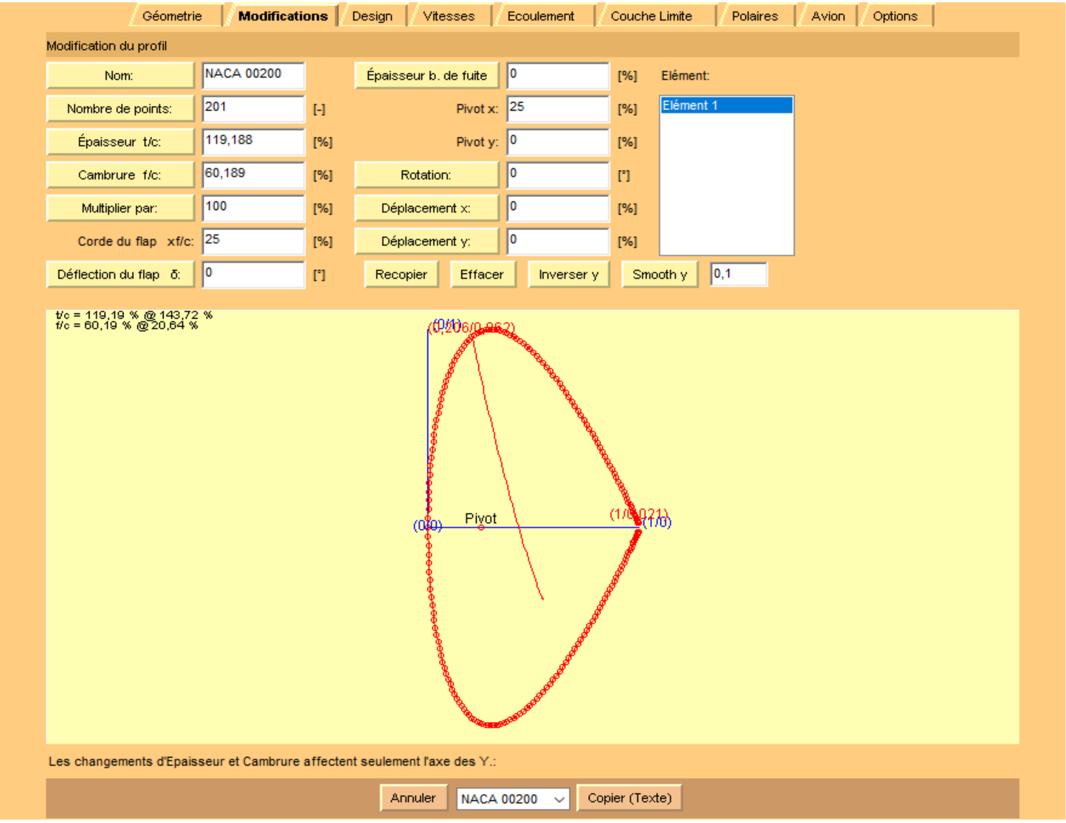
v2.79 | Verts:0 | Faces:0 | Tris:0 | Objects:0/2 | Lamps:0/1 | Mem:12.50M

Resolution: X: 1920 px, Y: 1080 px, 50%
Frame Range: Start Fra: 1, End Fr: 250, Frame St: 1
Aspect Ratio: X: 1.000, Y: 1.000
Frame Rate: 24 fps
Time Remappi... 10, 10
Anti-Aliasing: 5 8 11 16, Mitchell-Netr..., Full Sample, Si:1.000 px

*Figure 3.3- Python interface console (on Blender software).*

## 3.4 Java Foil

It is a relatively simple program, which uses several traditional methods of analysis. You just need to have some knowledge of the aerodynamic characteristics of an airplane wing profile. It then generates the necessary coordinates and then, through an analysis of the potential flow, highlights the pressure distribution around the profile as well as the lift and drag coefficients. It also allows you to vary several parameters such as the angle of attack and the speed of the flow by drawing the necessary curves [15].



Modification du profil

Nom: NACA 00200
Nombre de points: 201 [-]
Épaisseur t/c: 119,188 [%]
Cambrure f/c: 60,189 [%]
Multiplier par: 100 [%]
Corde du flap xf/c: 25 [%]
Déflection du flap δ: 0 [°]
Épaisseur b. de fuite 0 [%]
Pivot x: 25 [%]
Pivot y: 0 [%]
Rotation: 0 [°]
Déplacement x: 0 [%]
Déplacement y: 0 [%]
Smooth y 0,1
Elément: Elément 1

t/c = 119,19 % @ 143,72 %
f/c = 60,19 % @ 20,64 %

Les changements d'Epaisseur et Cambrure affectent seulement l'axe des Y.:

NACA 00200

*Figure 3.4- Java Foil interface.*

## 3.4 conclusion

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

.

# Chapter 4

# GUI REALIZATION

# Introduction

In this chapter, we will start with having some background about Python modules and, how Blender use them, then creating the Addon interface, and discuss the results.

## 4.1 Understanding Blender and Python

### ❖ The Default Blender Interface

When we first open up Blender, we get the familiar default user interface. We have a cube, a camera object, and a lamp object drawn into the scene shown in the 3D Viewport. Figure 4.1 shows the same interface with various major components labeled.

We discuss the function of each of these interfaces.

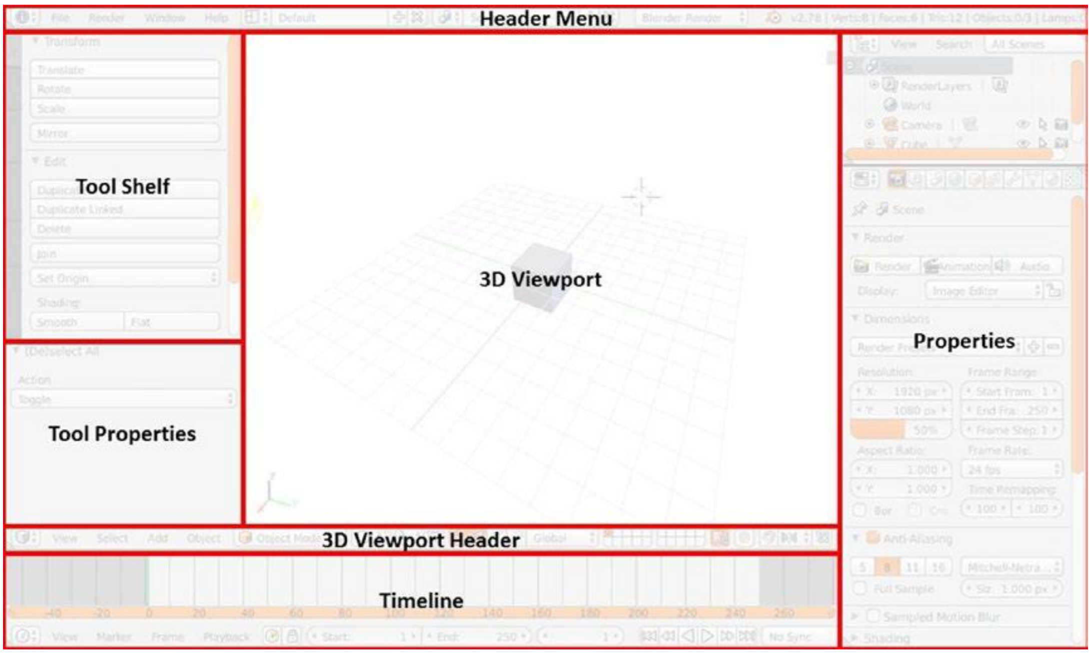


*Figure 4.1- The components of the Default Blender interface.*

### • 3D Viewport

The *3D Viewport*, or simply *Viewport*, gives us a preview of our work product. When we manipulate data in Blender, the 3D Viewport waits for all processes to finish writing data before updating itself. This is not noticeable in simple operations, like translations and rotations, that seem to happen instantaneously and in real time, but it is still important to acknowledge in add-on development.

The 3D Viewport has different viewing options and interaction options. Viewing options include *solid, wireframe,* and *rendered,* while interaction options include *Object Mode, Edit Mode,* and *Sculpt Mode [19]*.

### • Header Menu

The *Header menu* is a fairly standard header for a graphical user interface. It allows us to switch between interface layouts like *Default, Animation,* and *Scripting,* as well as switch between rendering engines like *Blender Render, Cycles Render,* and *Blender Game [19]*.

### • Properties Window

The *Properties* window allows us to access properties of objects, scenes, textures, animations, and more.

Most interfaces in the Properties window will give summaries and basic attributes rather than display all available details. It is very useful for keeping track of existing objects, object names, applied and unapplied transformations, and a few other important attributes. This window is generally always open in a Blender artist's layout, so it is a popular location to place add-on functions [19].

### • Tool Shelf and Tool Properties

The *Tool Shelf* is where different classes of operators are grouped by type. If we expand the window, we can see the Tool Shelf has various tabs like *Tools, Create,* and *Relations*. Most Blender add-ons will create a new tab in the Tool Shelf to hold its operators and parameters.

The *Tool Properties* window is a dynamic window that Blender populates with different sets of parameters depending on what tool the user has active. For example, when using the Rotate tool, we can fine-tune the rotation in this window instead of navigating to the exact spot in the Properties window that specifies rotation.

Tool Properties are advanced features typically intended to optimize ease-of-use rather than provide distinct functionalities to a tool. Many Blender add-ons ignore them altogether, and only a handful of native Blender tools use them [19].

- Timeline

The *Timeline* is used in animation. We can ignore this as we will not be animating [19].

## 4.2 Creating the Interface

In order to make our automated Modeling thesis, we need to create an interface contain tools that help us in drawing, controlling, and scaling the different dimensions of wing and winglet next, as an "Addon".

And here, we have begun in programming progress, Using Python language with the script and start creating our own program.

### 4.2.1 Addon Option

A Blender addon is just a piece of software that extend Blenders functionality. It can be in the form of a new tool, a new way to export a file format or a bunch of shortcuts that someone thought was a good idea to put together.

The goal of blender addons is generally the same: Simplify some process or add functionality [18].

### 4.2.2 Running Our Script

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 [31].

## ❖ The Scripting Interface

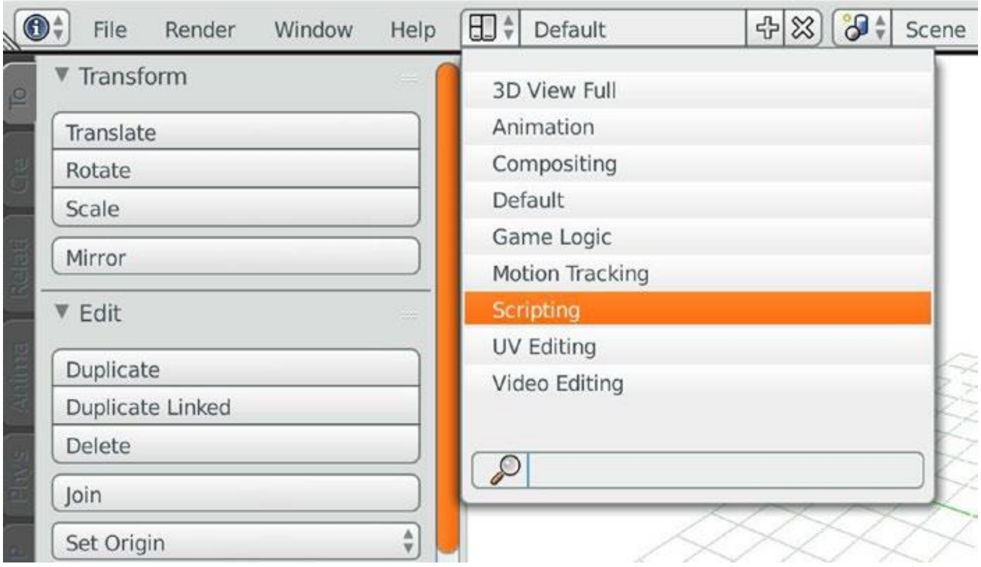
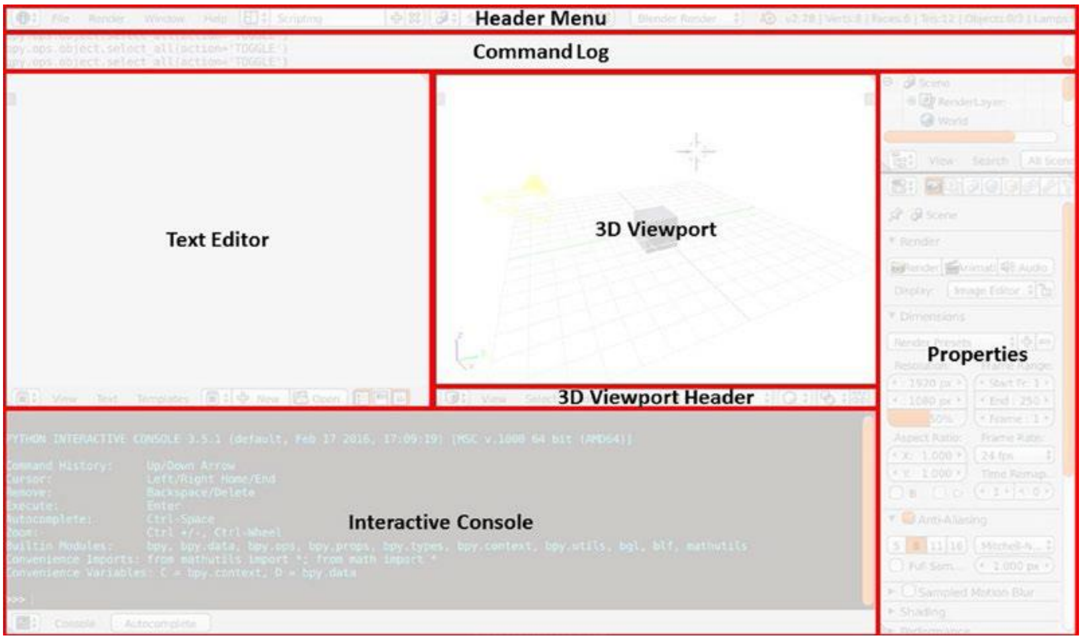


*Figure 4.2- Selecting the Scripting interface.*



```
bpy.ops.object.select_all(action='TOGGLE')
bpy.ops.object.select_all(action='TOGGLE')
```

```
PYTHON INTERACTIVE CONSOLE 3.5.1 (default, Feb 17 2016, 17:09:19) [MSC v.1800 64 bit (AMD64)]

Command History:     Up/Down Arrow
Cursor:              Left/Right Home/End
Remove:              Backspace/Delete
Execute:             Enter
Autocomplete:        Ctrl-Space
Zoom:                Ctrl +/-, Ctrl-Wheel
Builtin Modules:     bpy, bpy.data, bpy.ops, bpy.props, bpy.types, bpy.context, bpy.utils, bgl, blf, mathutils
Convenience Imports: from mathutils import *; from math import *
Convenience Variables: C = bpy.context, D = bpy.data

>>>
```

*Figure 4.3- Components of the Scripting interface.*

- ### Text Editor

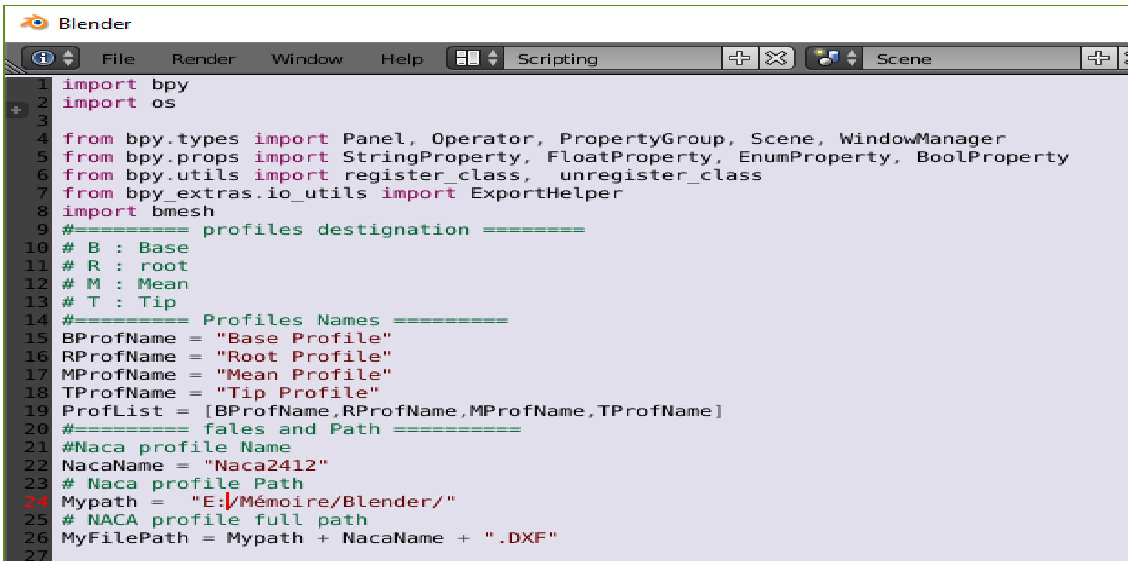
We can edit Python scripts (and any other text files) in the *Text Editor*. We can click the *New* and *Open* buttons to create and open scripts, respectively. Once scripts are loaded, the menu bar at the bottom of the Text Editor will change to allow saving and switching between files. Blender's Text Editor has some special properties pertaining to imports, system paths, and linked files in Python [19].

- ### Command Log

The *Command Log* shows function calls made by the Blender interface during the session. This window is extremely useful when experimenting with scripts and learning about the API [19].

- ### Interactive Console

The *Interactive Console* is a Python 3 environment similar to vanilla Python console and IPython consoles that often appear at the bottom of IDEs (interactive development environments). The Interactive Console does not share local or module-level data with the Text Editor scripts, but both Interactive Console and Text Editor scripts have access to the same global Blender data stored in bpy and its submodules. So, the console will not be able to read or modify variables local to the scripts, but modifications to bpy (and the Blender session in general) are shared [19].



```
import bpy
import os

from bpy.types import Panel, Operator, PropertyGroup, Scene, WindowManager
from bpy.props import StringProperty, FloatProperty, EnumProperty, BoolProperty
from bpy.utils import register_class,  unregister_class
from bpy_extras.io_utils import ExportHelper
import bmesh
#========== profiles destignation ========
# B : Base
# R : root
# M : Mean
# T : Tip
#========== Profiles Names =========
BProfName = "Base Profile"
RProfName = "Root Profile"
MProfName = "Mean Profile"
TProfName = "Tip Profile"
ProfList = [BProfName,RProfName,MProfName,TProfName]
#========== fales and Path ==========
#Naca profile Name
NacaName = "Naca2412"
# Naca profile Path
Mypath =  "E:/Mémoire/Blender/"
# NACA profile full path
MyFilePath = Mypath + NacaName + ".DXF"
```

*Figure 4.4- Modules and submodules used.*

## • Modules and Functions Overview

We begin by giving some background about modules we used.

### A. 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].

## B. The bmesh Module

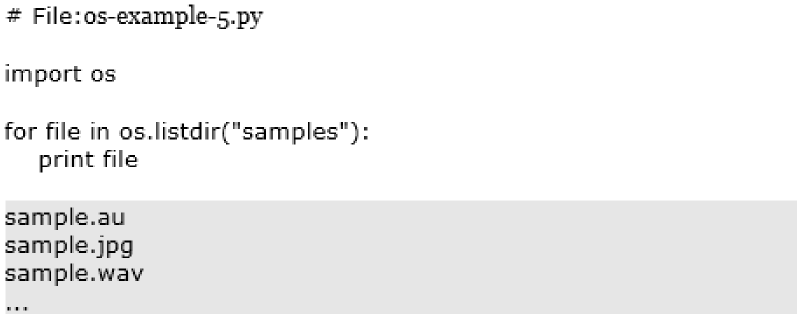
In Blender, bmesh objects are fairly heavy-handed and computationally expensive when compared to other core data structures. To maintain efficiency, Blender gives much of the data and instance management work to the user to manage via the API.

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].

## C. 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].



```
# File:os-example-5.py

import os

for file in os.listdir("samples"):
    print file
```

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

Figure 4.5: Example of Using the OS module to list the files in a directory.

## D. Function

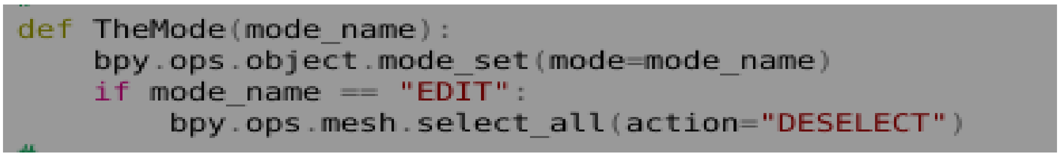
A function is a block of code which only runs when it is called.

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 4.6- Example of Function.*

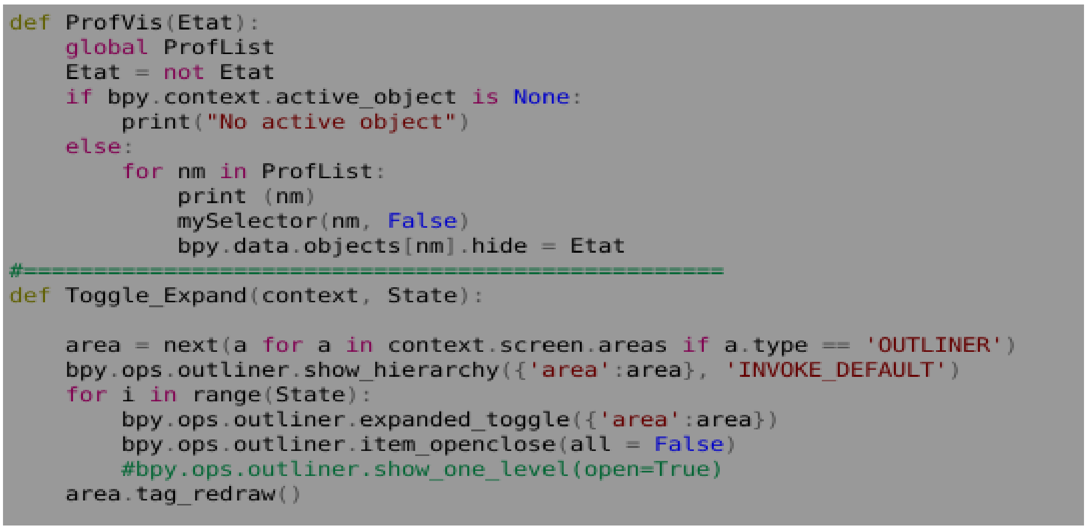
## E. Booleans

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 4.7- Example of Booleans.*

## 4.3 profiles Realization

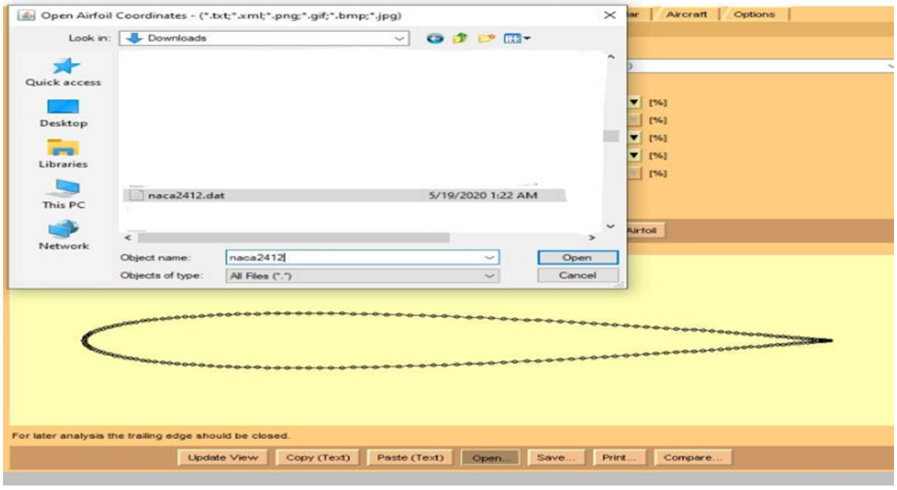
- DXF format

AutoCAD DXF (Drawing Interchange Format, or Drawing Exchange Format) is a CAD data file format developed by Autodesk for enabling data interoperability between AutoCAD and other programs [35].

- NACA Airfoil

Using “Java Foil” software we have generated the 200 coordinate points of our NACA 2412 profile used simultaneously for the wing and for the winglet.

– Download the data file of the airfoil required from “UIUC Airfoil Data Site”.
– Open the "Java Foil " software.
– Choose the “open” option and open the downloaded data file to save the airfoil with “.dxf” extension.



Look in: Downloads

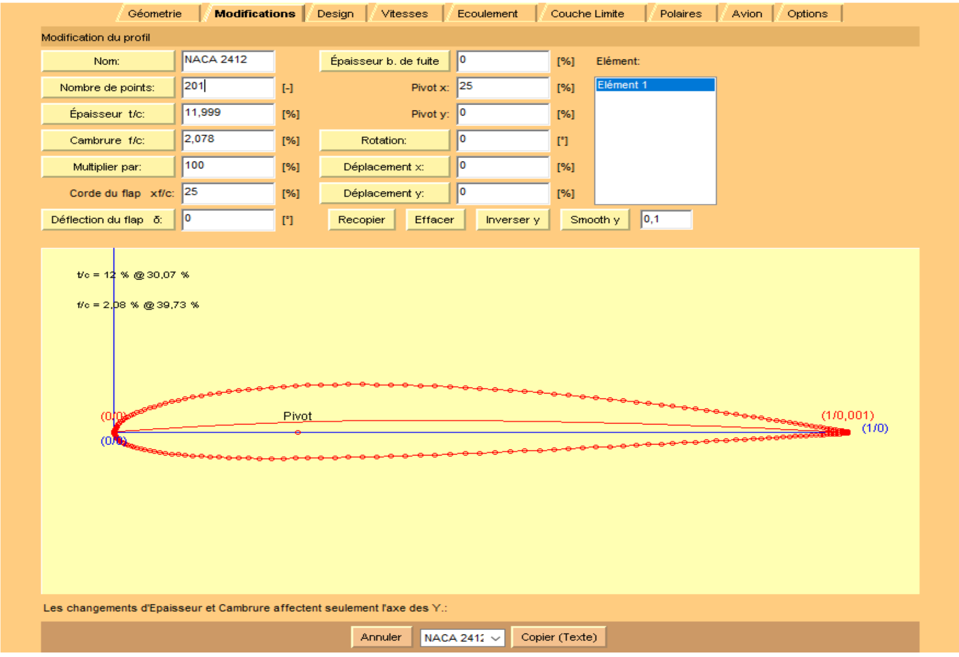
naca2412.dat 5/19/2020 1:22 AM

Object name: naca2412

Objects of type: All Files (*.*)

For later analysis the trailing edge should be closed.

*Figure 4.8- Opening the data file (Java Foil software).*



Modification du profil

| | | | | | |
|---|---|---|---|---|---|
| Nom: | NACA 2412 | | Épaisseur b. de fuite | 0 | [%] |
| Nombre de points: | 201 | [-] | Pivot x: | 25 | [%] |
| Épaisseur t/c: | 11,999 | [%] | Pivot y: | 0 | [%] |
| Cambrure f/c: | 2,078 | [%] | Rotation: | 0 | [°] |
| Multiplier par: | 100 | [%] | Déplacement x: | 0 | [%] |
| Corde du flap xf/c: | 25 | [%] | Déplacement y: | 0 | [%] |
| Déflection du flap δ: | 0 | [°] | Smooth y | 0,1 | |

Elément: Elément 1

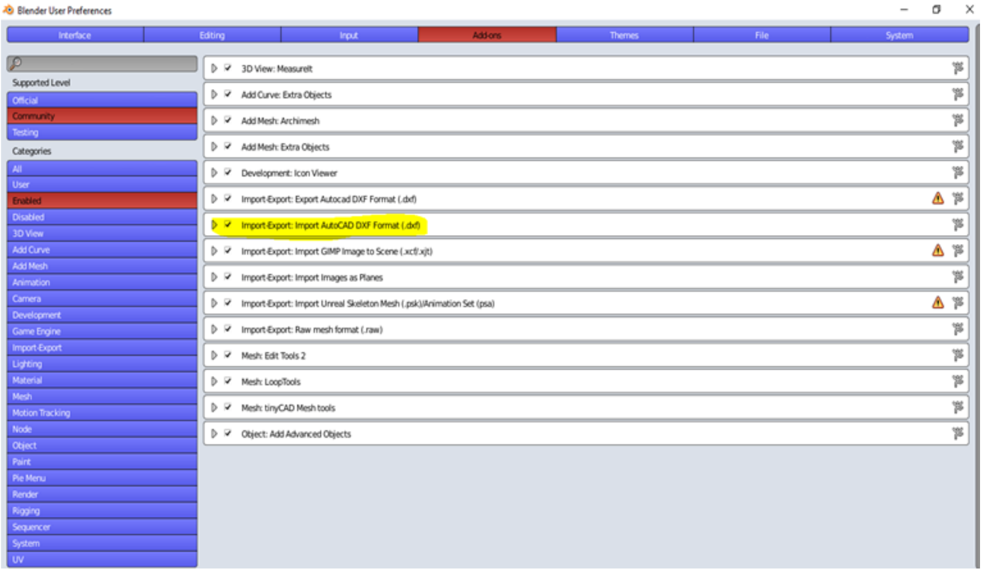
t/c = 12 % @ 30,07 %

f/c = 2,08 % @ 39,73 %

Les changements d'Epaisseur et Cambrure affectent seulement l'axe des Y.:

NACA 2412

*Figure 4.9- Profile NACA 2412 (Java Foil software).*

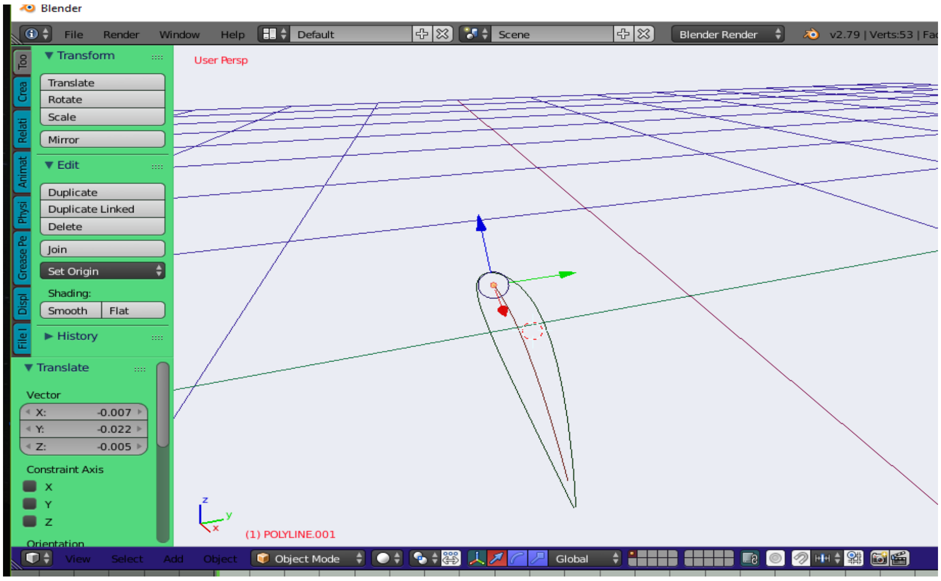


- ☑ 3D View: MeasureIt
- ☑ Add Curve: Extra Objects
- ☑ Add Mesh: Archimesh
- ☑ Add Mesh: Extra Objects
- ☑ Development: Icon Viewer
- ☑ Import-Export: Export Autocad DXF Format (.dxf)
- ☑ Import-Export: Import AutoCAD DXF Format (.dxf)
- ☑ Import-Export: Import GIMP Image to Scene (.xcf/.xjt)
- ☑ Import-Export: Import Images as Planes
- ☑ Import-Export: Import Unreal Skeleton Mesh (.psk)/Animation Set (psa)
- ☑ Import-Export: Raw mesh format (.raw)
- ☑ Mesh: Edit Tools 2
- ☑ Mesh: LoopTools
- ☑ Mesh: tinyCAD Mesh tools
- ☑ Object: Add Advanced Objects

*Figure 4.10- The Add-ons options in User Preference (Blender).*

We have activated the option of “Import Export: Import DXF Format (.dxf)”, Then import profile (object mode).

- Open the airfoil dxf file in Blender: File➤ import ➤ AutoCAD DXF



Translate

Vector
X: -0.007
Y: -0.022
Z: -0.005

Constraint Axis
X
Y
Z

Orientation

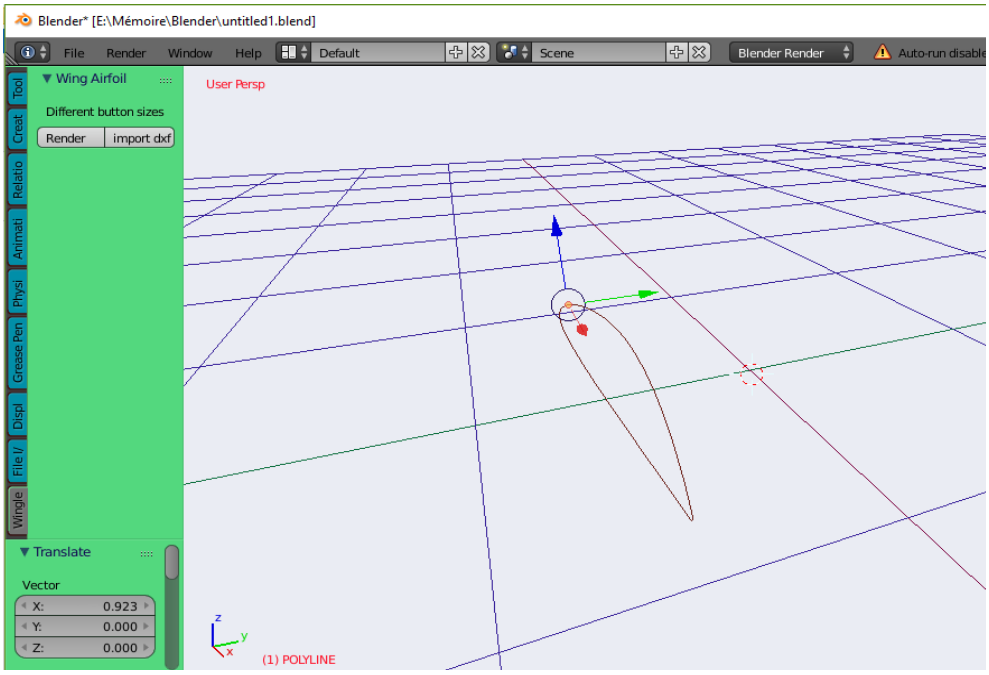
*Figure 4.11- Importing NACA2412 profile (Blander).*

## 4.4 Importing Profiles

After, figuring out which function adds a DXF profile to the scene. Navigate to the “Menu Header” and go to 3D Viewport Header ➤ Import AutoCAD DXF ➤ dxf file, then navigate to the Command Log to verify that the function was executed which “bpy.ops.import_scene.dxf”

Using the submodule “bpy.ops.import_scene.dxf” we created have a simple operator to import DXF profile, and do the following changes in the function that was executed:

- The scale from 1 to 0.01.
- Use the Booleans to unmerge the merged object.
- Delete the selected object “POLYLINE 02”.
- Determined the file path



Vector

X: 0.923

Y: 0.000

Z: 0.000

*Figure 4.12- Importing the final Used NACA2412 profile (Blander).*

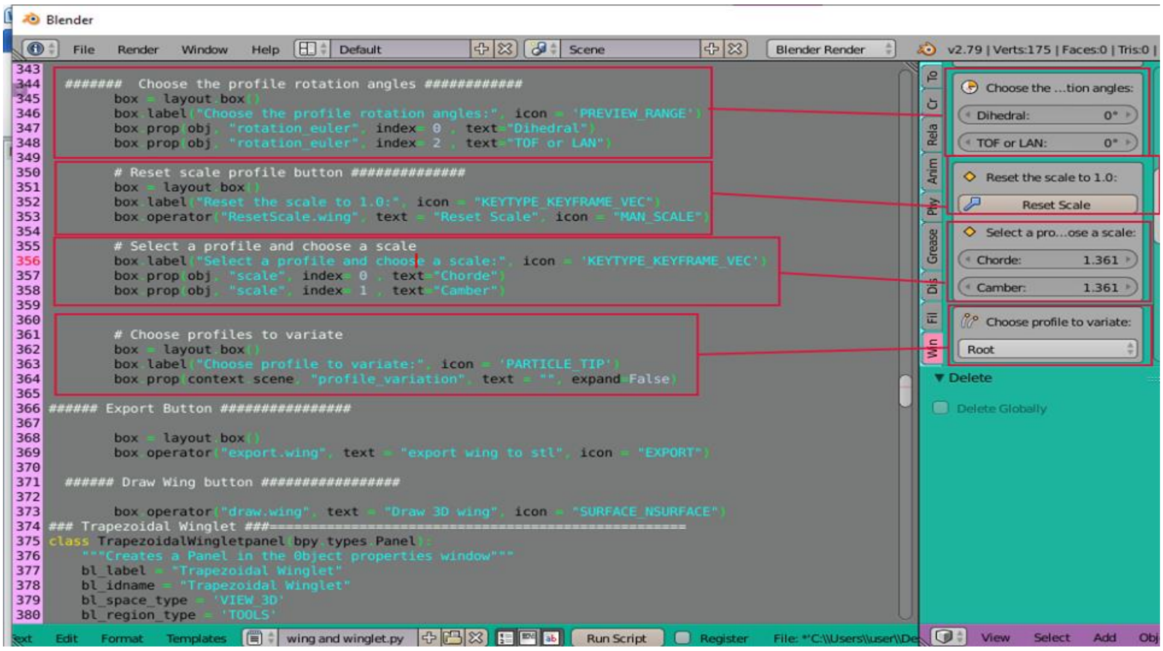
## Note:

To find functions, classes, parameter lists, and minimal documentation, use the autocomplete feature

of Blender's Interactive Console. With the mouse cursor in the window of the Interactive Console, begin typing a bpy function. Press **Ctrl+Space**, and Blender will show class and function information.

## 4.5 Manipulation of profiles

- Operators



```
####### Choose the profile rotation angles ############
        box = layout.box()
        box.label("Choose the profile rotation angles:", icon = 'PREVIEW_RANGE')
        box.prop(obj, "rotation_euler", index= 0 , text="Dihedral")
        box.prop(obj, "rotation_euler", index= 2 , text="TOF or LAN")

        # Reset scale profile button ##############
        box = layout.box()
        box.label("Reset the scale to 1.0:", icon = "KEYTYPE_KEYFRAME_VEC")
        box.operator("ResetScale.wing", text = "Reset Scale", icon = "MAN_SCALE")

        # Select a profile and choose a scale
        box.label("Select a profile and choose a scale:", icon = 'KEYTYPE_KEYFRAME_VEC')
        box.prop(obj, "scale", index= 0 , text="Chorde")
        box.prop(obj, "scale", index= 1 , text="Camber")


        # Choose profiles to variate
        box = layout.box()
        box.label("Choose profile to variate:", icon = 'PARTICLE_TIP')
        box.prop(context.scene, "profile_variation", text = "", expand=False)

###### Export Button ################

        box = layout.box()
        box.operator("export.wing", text = "export wing to stl", icon = "EXPORT")

   ###### Draw Wing button #################

        box.operator("draw.wing", text = "Draw 3D wing", icon = "SURFACE_NSURFACE")
### Trapezoidal Winglet ###==================================================
class TrapezoidalWingletpanel(bpy.types.Panel):
    """Creates a Panel in the Object properties window"""
    bl_label = "Trapezoidal Winglet"
    bl_idname = "Trapezoidal Winglet"
    bl_space_type = 'VIEW_3D'
    bl_region_type = 'TOOLS'
```

Choose the ...tion angles:
Dihedral: 0°
TOF or LAN: 0°

Reset the scale to 1.0:
Reset Scale

Select a pro...ose a scale:
Chorde: 1.361
Camber: 1.361

Choose profile to variate:
Root

▼ Delete
□ Delete Globally

*Figure 4.13- Operators (Blender scripting interface).*

- Initial Profiles Chords

– Chord B = 9.0111

– Chord R = 7.6042

– Chord M = 3.8100

– Chord T = 1.3610

**Span:** it is the distances between two profiles according to Z.

- Initial Span Profiles Positions

– ZB = 0.0000

– ZR = 0.7035

– ZM = 6.2035

– ZT = 17.7035

– ZMAX = 20.0000

**Axial distance:** it is the distance between two profiles according to X.

- Initial Axial Profiles Positions

- $X_B$ = 0.0000
- $X_R$ = 1.4069
- $X_M$ = 5.2011
- $X_T$ = 7.6501
- $X_{MAX}$ = 30.0000

**Relative distances**: it is the distance between the span and axial.

- Initial relative distances (Interesting for design)

- Dz BR = 0.7035 Span distance between B and R
- Dz RM =5.5000 Span distance between R and M
- Dz MT =11.5000 Span distance between M and T
- Dx BR = 1.4069 Axial distance between B and R
- Dx BM = 3.7942 Axial distance between R and M
- Dx MT = 2.4490 Axial distance between M and T

**Dihedral:** In aeronautics, dihedral is 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 and the upward angle of an aircraft's wings [33].

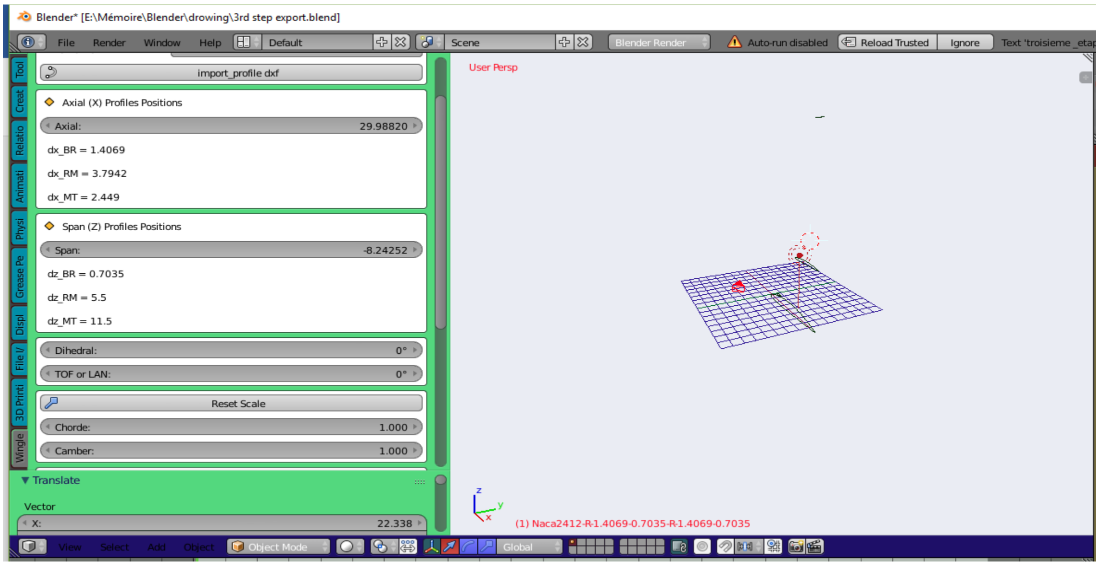
**Angle of attack:** In aerodynamics, angle of attack specifies the angle between the chord line of the wing of a fixed-wing aircraft and the vector representing the relative motion between the aircraft and the atmosphere [34].

- Dihedral angle and Angle of attack (TOF or LAN)

- Dihedral angle = 0°
- Angle of attack (TOF or LAN) = 0°

## Note:

we left the option of modifying in those dimensions open according to each case, you could change in the dimension you want to change by clicking on the specific button that was created by the operators we used. The figure bellow shows each dimension button.



Blender* [E:\Mémoire\Blender\drowing\3rd step export.blend]

Axial (X) Profiles Positions

Axial: 29.98820

dx_BR = 1.4069

dx_RM = 3.7942

dx_MT = 2.449

Span (Z) Profiles Positions

Span: -8.24252

dz_BR = 0.7035

dz_RM = 5.5

dz_MT = 11.5

Dihedral: 0°

TOF or LAN: 0°

Chorde: 1.000

Camber: 1.000

Translate

Vector

X: 22.338

(1) Naca2412-R-1.4069-0.7035-R-1.4069-0.7035

*Figure 4.14- Dimensions button (Blender).*

## 4.6 3D Drawing of the wing

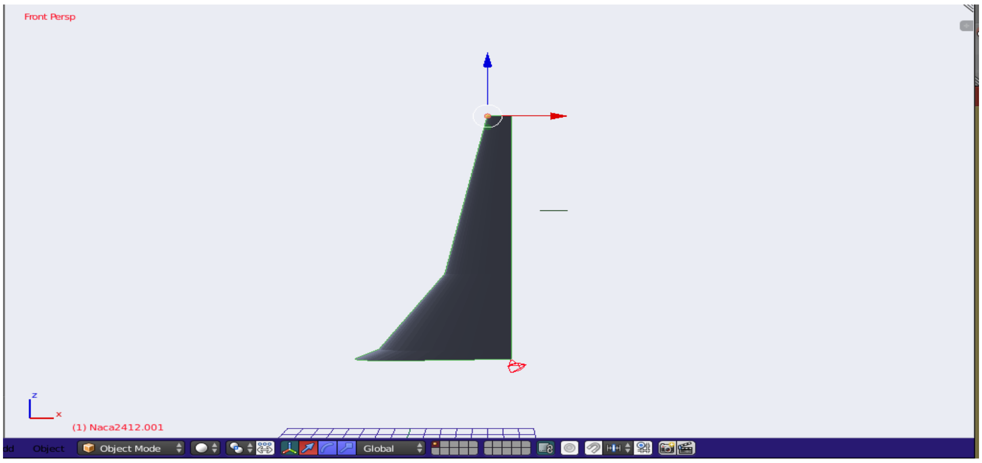


*Figure 4.15- The front view of the wing (Blender).*

## 4.7 Wing exportation

As we worked previously with DXF format, using the operator shown in figure (4.17) we created a button to export the wing (same with winglet).

we have used STL format for the exportation of the two objects (wing and winglet).

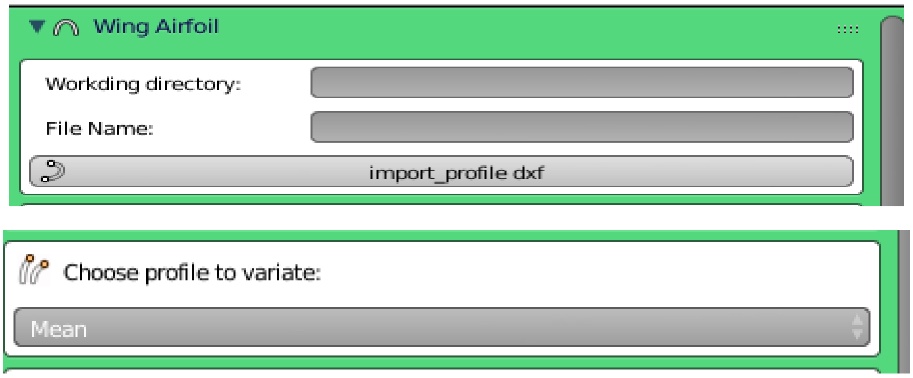
- STL format

STL (an abbreviation of "stereolithography") is a file format native to the stereolithography CAD software created by 3D Systems. STL has several backronyms such as "Standard Triangle Language" and "Standard Tessellation Language". This file format is supported by many other software packages; it is widely used for rapid prototyping, 3D printing and computer-aided manufacturing. STL files describe only the surface geometry of a three-dimensional object without any representation of color, texture or other common CAD model attributes. The STL format specifies both ASCII and binary representations. Binary files are more common, since they are more compact [36].

- How the export file is saved?

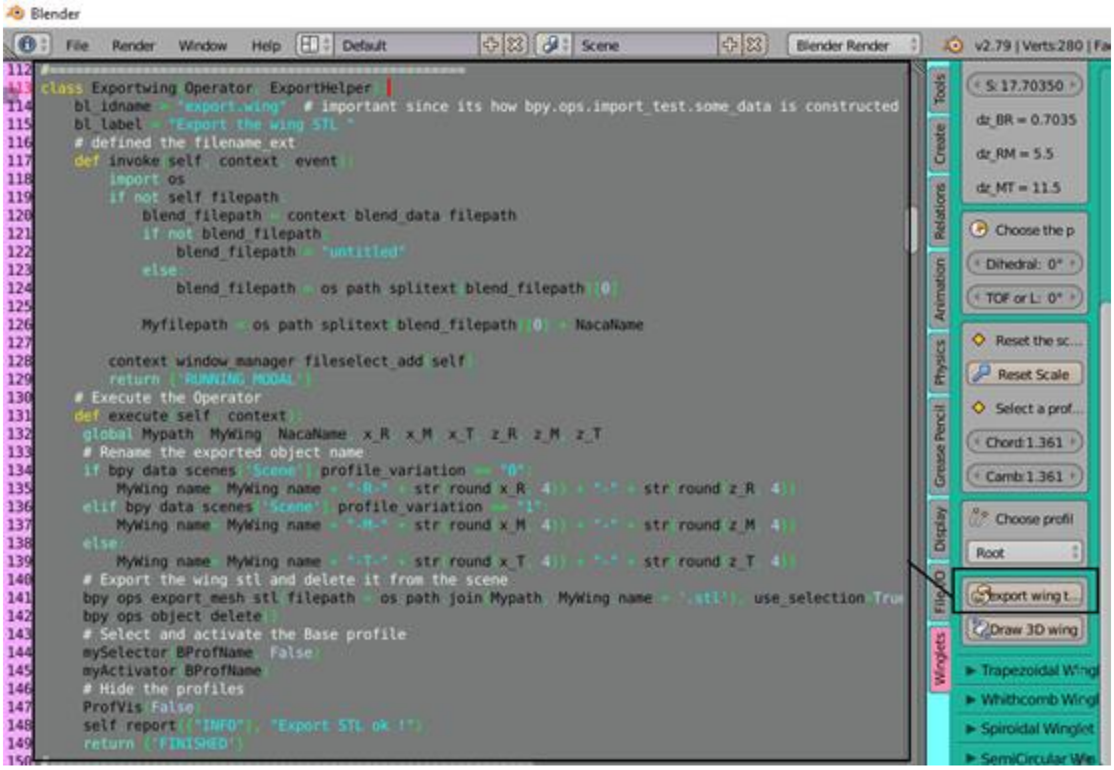
In order to not cause a confusion when we make changes in dimensions, we made this feature, which enable us to choose the desired profile (mean, tip, or root) we want to change its dimensions, and give the object a new name when exporting it.

You could also determine the file by giving the file a name using the “file name” feature.



Choose profile to variate: Mean

*Figure 4.16- File name feature.*



```
#===================================================
class Exportwing(Operator, ExportHelper):
    bl_idname = "export.wing"  # important since its how bpy.ops.import_test.some_data is constructed
    bl_label = "Export the wing STL "
    # defined the filename_ext
    def invoke(self, context, event):
        import os
        if not self.filepath:
            blend_filepath = context.blend_data.filepath
            if not blend_filepath:
                blend_filepath = "untitled"
            else:
                blend_filepath = os.path.splitext(blend_filepath)[0]

            Myfilepath = os.path.splitext(blend_filepath)[0] + NacaName

        context.window_manager.fileselect_add(self)
        return {'RUNNING_MODAL'}
    # Execute the Operator
    def execute(self, context):
     global Mypath, MyWing, NacaName, x_R, x_M, x_T, z_R, z_M, z_T
     # Rename the exported object name
     if bpy.data.scenes['Scene'].profile_variation == "0":
         MyWing.name= MyWing.name + "-R-" + str(round(x_R, 4)) + "-" + str(round(z_R, 4))
     elif bpy.data.scenes['Scene'].profile_variation == "1":
         MyWing.name= MyWing.name + "-M-" + str(round(x_M, 4)) + "-" + str(round(z_M, 4))
     else:
         MyWing.name= MyWing.name + "-T-" + str(round(x_T, 4)) + "-" + str(round(z_T, 4))
     # Export the wing stl and delete it from the scene
     bpy.ops.export_mesh.stl(filepath = os.path.join(Mypath, MyWing.name + '.stl'), use_selection=Tru
     bpy.ops.object.delete()
     # Select and activate the Base profile
     mySelector(BProfName, False)
     myActivator(BProfName)
     # Hide the profiles
     ProfVis(False)
     self.report({"INFO"}, "Export STL ok !")
     return {'FINISHED'}
```

*Figure 4.17- Exportation button (Blender).*

## 4.8 3D Drawing of the Winglet

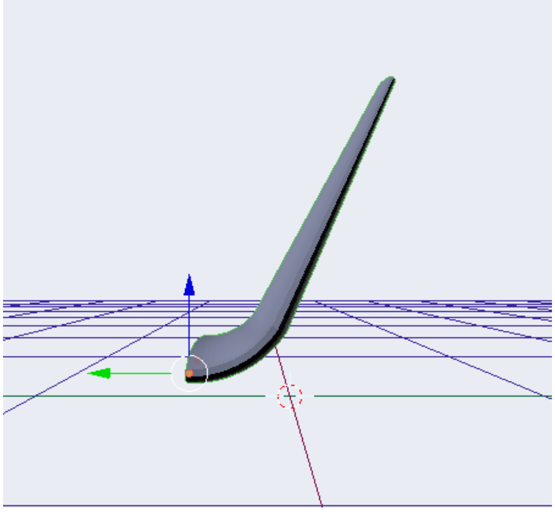


*Figure 4.18- left view of Blended winglet.*

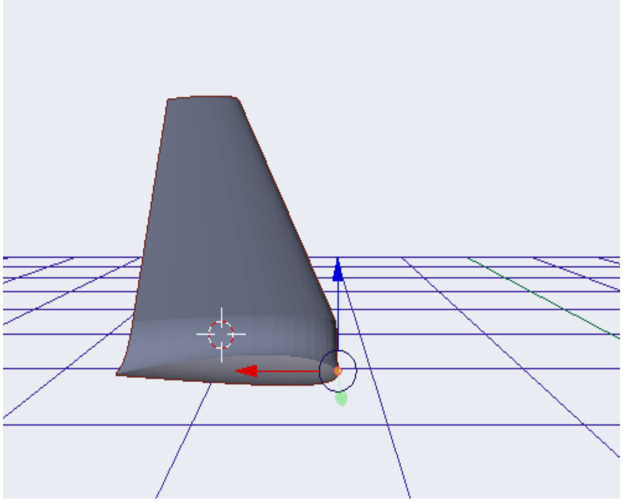


*Figure 4.19- Back view of Blended winglet (Blender).*

## 4.9 Conclusion

Considering the means at our disposal, we can conclude that we reached to the result we wanted and successfully build the graphical user interface.

# General conclusion and perspectives

In this study, we first understood the difficulties in conception of such a complex geometry (Boeing 757-200, and winglets). We have understood and learned the steps to follow to complete both the conception and the graphical user interface. It should be taking in 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. We have 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 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 concepting the 257-200aircraft wing and winglet on Blender, finally work on the graphical interface, that was successfully build, 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 ability to modify in the dimensions design, through this interface we are ascertaining the best winglet performance and improving the efficiency of wing.

As perspectives to this work, we recommend that Blended winglet model should be made of 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 design that were created from this interface when the aircraft is flying at cruise velocity.

# References

[1] KENZA LAALAOUI, "Amélioration de l'écoulement autour d'une aile d'avion par l'utilisation de winglet de type trapézoidale" 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] Amador Calvo Portillo, "CFD ANALYSIS OF WINGLETS", GLYNDWE university WREXHAM, Aeronautical & Mechanical Engineering, May 2011.

[4] Winglets, NASA technology facts, NASA Dryden Flight Research Center 2004.

[5] https://airlineworld.wordpress.com/2008/10/01/aircraft-winglets/ ,"Aircraft Winglets", October 1, 2008.

[6] https://en.wikipedia.org/wiki/Oswald_efficiency_number , Oswald efficiency number, 25 June 2017.

[7] www.innovationpei.com/ebusiness " Open Source Software", INNOVATION PEI.

[8] Sukhwinder Randhawa, "Open Source Software and Libraries", January 2008.

[9] James Chronister, "BlenderBasics",4thEdition,2011.

[10] https://docs.blender.org/manual/en/latest/getting_started/about/introduction.html, About Blender, Introduction.

[11] Prof. Dr. Eng. Tihomir Dovramadjiev, "1. Introduction to Blender software. Interface", Department Industrial Design, Technical University of Varna, September 2018.

[12] Prof. Dr. Eng. Tihomir Dovramadjiev, "2 Introduction to Blender software. 3D View Editors", Department Industrial Design, Technical University of Varna, September 2018.

[13] https://www.python.org/doc/essays/blurb/

[14] https://python.sdv.univ-paris-diderot.fr/ Patrick Fuchs and Pierre Poulain, « Cours de Python », Paris university, 22 juillet 2020.

[15] http://www.mh-aerotools.de/airfoils/javafoil.htm , JavaProp - Design and Analysis of Propellers.

[16] Gianluca Minnella, Yuniesky Rodriguez, Jose Uga, “Aerodynamic Shape Design Optimization of Winglets”, Department of Mechanical Engineering, FLORIDA International University, October 25, 2010.

[17]https://www.boeing.com/resources/boeingdotcom/company/about_bca/startup/pdf/historical/757_passenger.pdf « 757_passenger », Startup Boeing, 2007

[18] https://artisticrender.com/30-best-blender-addons-install-manage-free-and-paid/ , Erik Selin, “30 Blender addons (install, manage, free and paid)”.

[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://www.w3schools.com/python/python_functions.asp , Python functions.

[24] https://www.w3schools.com/python/python_booleans.asp , Python Booleans

[25] Dr. Laïd MESSAOUDI, « MECANIQUE DES FLUIDES INCOMPRESSIBLES », Université Batna 2, 2020.

[26] Saravanan 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] Dieter SCHOLZ, “Definition and discussion of the intrinsic efficiency of winglets”, Aircraft Design and Systems Group (AERO), Hamburg University of Applied Sciences, 16th-20th October 2017.

[28] R. T. Whitcomb, A Design Approach and Selected Wind-Tunnel Results at High Subsonic Speeds for Wingtip Mounted Winglets, NASA-TN-D-8260, Report No. L-10908, NASA Langley Research Center, Hampton, VA 23665, USA, URL: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19760019075.pdf, 1976.

[29] https://conself.com/blog/calculate-lift-drag-with-paraview/ , Calculate Lift & Drag with Paraview, 13 February 2020.

[30] Vinay Kumar Bada[1], Kumari Monika[2], Akram Hussain.Md[3], Praveenkumar.chikoti[4], "CFD Analysis and Comparison of Spiroid and Dual Feather Winglets" Department of Aeronautical Engineering, Vardhaman College of Engineering, 1 August 2016.

[31] https://docs.blender.org/api/2.79/info_quickstart.html

[32]https://people.clarkson.edu/~pmarzocc/AE429/The%20NACA%20airfoil%20series.pdf , The NACA Airfoil Series.

[33] https://en.wikipedia.org/wiki/Dihedral_(aeronautics), Dihedral (aeronautics), 28 May 2020.

[34]https://en.wikipedia.org/wiki/Angle_of_attack, Angle of attack, 19 August 2020.

[35] https://en.wikipedia.org/wiki/AutoCAD_DXF , AutoCAD DXF, 23 June 2020.

[36] https://en.wikipedia.org/wiki/STL_(file_format) , STL (file format), 1 September 2020.