TÉCNICO LISBOA

Aircraft Optimal Design

Aerospace Engineering

José Miguel Fonseca Santos

83704

4th Assignment

08/12/2020

Contents

1 Introduction

Nowadays there is a large number of high-fidelity optimization softwares available, however, it is common to for aicraft design teams to begin their analyses with low-fidelity tools and move on to higher-fidelity options at later stages. In this report, a low-fidelity resource will be used in order to solve optimization problems. The software used is OpenAeroStruct, which is a free tool that allows aerostructural optimization combining a vortex-lattice method (VLM) and a finite element method to perform the simulation of aerodynamic and structural analyses using lifting surfaces. This report will cover the steps taken in the optimization process of a wing and tail configuration.

2 Optimization problem formulation

The optimization problem chosen will be based on the Bombardier CRJ700. With the aid of OpenAeroStruct, an aircraft aerostructural design will be conducted, using the FEM tube model. The optimization problem will have the the following structure:

$$\begin{cases} minimize \quad \texttt{fuelburn} \\ w.r.t. \quad \texttt{wing twist}, \texttt{angle of attack}, \texttt{wing\_thickness} \\ subjet \quad to \quad C\_Lcruise = 0.47, L_{maneuver} = W_{maneuver}, \sigma_{VonMises} \geq \sigma_{lim} \\ \quad , \texttt{twist at root} = 0 \end{cases} \tag{1}$$

The objective will be to reduce the fuelburn on account of changing wing parameters that will improve the flight conditions subject to analyses and thus reduce the fuel required. The fuel (f) is calculated based on the Brequet equation defined by:

$$f = (W_0 + W_S)e^{\frac{R*CT*C_D}{a*M*C_L}} \tag{2}$$

In this equation W_0 stands for the operating empty weight of the aircraft, without fuel or structural mass. W_S represents the structural weight, a the speed of sound, M the flight Mach number and R the range. The values used to compute the fuel will correspond to a cruise flight. Since it is required to perform an aerostructural optimization, e.g. relate the aerodynamic (VLM) with the structural (FEM) analyses. This way, it is a good practice to define multiple points on the problem. The first point corresponds to cruise conditions, while the second point will correspond to a maneuver at sea level. The main reason to define multiple points is to allow the constraints to be better defined. During a maneuver, the stresses at which the wing is subjected are higher than at cruise condition. Thus, the structural analyses will be performed taking into account the maneuver, ensuring that the aircraft will not only resist the bending moments and shear loads at cruise, but also while being subjected to an higher load factor. The optimizer used will be ScipyOptimizeDriver default, which is the 'SLSQP'. Some of the default values for this method are shown in the following table. The tolerance corresponds to the algorithm termination tolerance.

Parameters	**Default**
Algorithm	'SLSQP'
MaxIterations	200
Tolerance	1×10^{-6}

Table 1: Default values for ScipyOptimizeDriver

Before presenting some results, it is important to take a look at some of the values used to defined some parameters that helped to get the first optimization results.

Variable	**Value**	**Units**
num_x	17	-
num_y	3	-
span	23.24	m
taper_ratio	0.3	
C_{L_0}	0.1	
C_{D_0}	0.0078	
$\frac{c}{t}_{max}$	12%	
chord for $\frac{c}{t}_{max}$	0.37	
Young's modulus	71.7×10^9	Pa
Shear modulus	26.9×10^9	Pa
Max. yield stress	$503 \times 10^6/1.5$	MPa
Density	2810	kg/m^3
wing_weight_ratio	1.25	

Table 2: Wing parameters

Variable	**Value**	**Units**
num_x	17	-
num_y	3	-
span	8	m
C_{L_0}	0.0	-
C_{D_0}	0.0078	-
chord for $\frac{c}{t}_{max}$	0.30	
Young's modulus	71.7×10^9	Pa
Shear modulus	26.9×10^9	Pa
Max. yield stress	$503 \times 10^6/1.5$	MPa
Density	2810	kg/m^3
wing_weight_ratio	1.2	

Table 3: Tail parameters

Some of these parameters are further changed in order to understand their influence. The structural values were obtained through some research on the alloy Al 7075-T6.

As previously mentioned before, the optimization will be mainly constituted by two flight condition points. It is also adequate to define each flight condition before analysing the results. The cruise altitude is 35.000 feet, and the following values are obtained according to the U.S Standard Atmosphere Data [2]:

$$\rho = 0,380 kg/m^3 \qquad T = 220,6K \qquad \mu = 1,434 * 10^{-5} Ns/m^2$$

For the operating conditions, both the cruise velocity and Reynolds number divided by the chord are easily obtained:

$$a = \sqrt{\gamma * R * T} == 297.7 \quad m/s$$
$$V = M * a = 232.2 \quad m/s$$
$$\frac{Re}{c} = \frac{\rho * v}{\mu} = 6 * 10^6 \quad 1/m$$

The percentage of the chord in laminar flow is assumed to have a small value of 5 percent since the Reynolds number has a large order of magnitude. The flight conditions are thus shown in the following tables.

Variable	Value	Units
Altitude	35000	ft
Mach	0.78	-
Cruise velocity	232.2	m/s
Air density	0.380	kg/m^3
Sound speed	297.7	m/s
SFC	0.38	lb/(lb h)
Range	3121	km
Load factor	1.0	-

Table 4: Cruise condition

Variable	Value	Units
Altitude	0	ft
Mach	0.6	-
Air density	1.225	kg/m^3
Sound speed	$340.$	m/s
Load factor	2.5	-

Table 5: Maneuver condition

2.1 N2 diagram

Having defined the most important parameters, it is now possible to follow the optimization procedure and engage in the analyses of the N2 diagram. The N2 diagram helps the user to have a better understanding of how the variables are being transferred from one module to the other. It is also crucial to identify the name of the pointer to each variable, which is then used by the software and make the required connections. This tool is useful throughout optimization problems to verify the logistic of the code and correct errors that may appear due to the disconnection between variables. One problem that appears related to the N2 diagram is the difficulty to to analyse it once the optimization problem starts to get complex. The N2 diagram shown above is collapse in order to only present major groups, hiding several components.

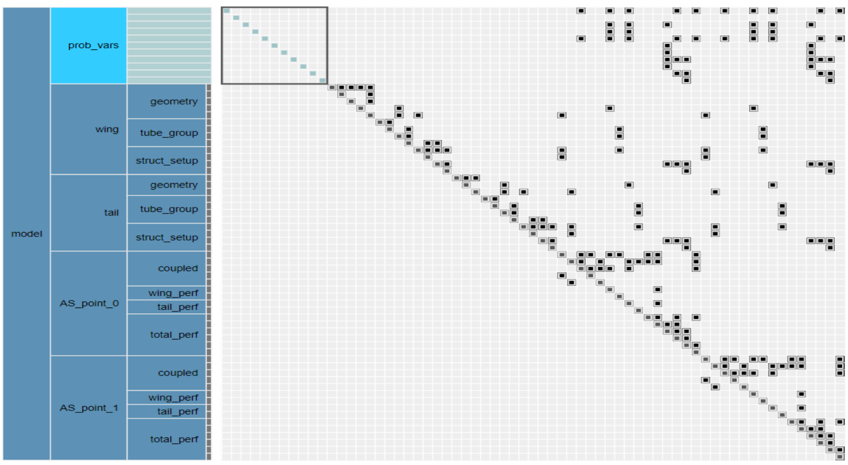

Figure 1: N2 diagram overview

In this N2 diagram it is easy to identify the independent problem variables added, such as velocity, Mach number, density, amongst others. There are 4 major groups, the first two contain geometric information regarding the wing and tail mesh. The second two, *AS_point_0* and *AS_point_1*, contain the information regarding the aerodynamic and structural analyses on both flight conditions. The cruise condition is associated to *AS_point_0* and the maneuver to the *AS_point_1*. One step that is highly recommended to take before running the driver is to take some time zooming in and out and look for variables colored in red. The color red represents that the respective variable is not getting information from any other component or from the user. When this happens, some variables will take default values, which depending on the variable may or may not be harmful to the final solution.

2.2 First optimization

In the first optimization performed, the problem is defined in the following form:

$$\begin{cases} minimize \quad \texttt{fuelburn} \\ w.r.t. \quad \texttt{wing\_twist@cp}, \texttt{angle\_of\_attack}, \texttt{wing\_thickness@cp} \\ subjet \quad to \quad C_{L_{cruise}} = 0.47, \texttt{twist @root} = 0 \\ \qquad \sigma_{VonMises} \geq \sigma_{lim} L_{maneuver} = W_{maneuver} \end{cases} \tag{3}$$

The program solved the problem successfully, since the exit mode is 0. The function value decreased to the order of $\times 10^-2$, which is also a good sign. As for the number of iterations, function evaluations and gradient evaluations, despite being at an early stage, is assumed to be reasonable low. When comparing these values to the ones obtained in previous reports, and also comparing the complexity of the problem, it can be said that the computer cost associated with solving this problem in OpenAeroStruct is quite low.

Parameter	**Value**
Exit mode	0
Current function value	0.02011
Iterations	75
Function evaluations	78
Gradient evaluations	75

Table 6: Optimization parameters

The results obtained at a first impression are positive, since it is still the first iteration. The lift distribution follows almost perfectly the elliptical lift distribution. Also, one of the constraints, e.g. $C_{L_{cruise}} = 0.47$, ensures that the lift coefficient in cruise is adequate. This value is easily calculated having already an estimate for the aircraft weight and cruise altitude. As for the structural analyses, one can conclude that the stress is higher at the root part of the wing, although it never reaches the structural limit imposed. One reason for this fact is that the wing chord is not constrained and having a bigger value for chord will distribute the stress and increase resistance. This gives a hint for the next step in the iteration process. The thickness of the wing is constant over the span and equal to the minimum used to constrain it due to the fact that the failure limit is not attained and thus it is not required to increase the thickness in order to support higher forces.

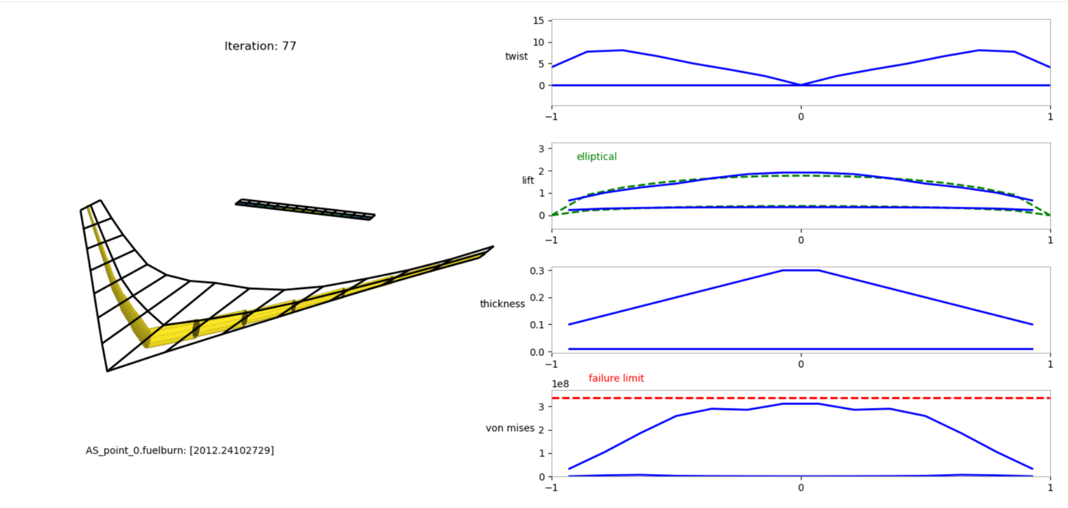

Figure 2: Results of first iteration (*plot_wing* script)

Parameter	**Value**
`alpha`	7.88
`wing_twist_cp`	$(4.12, 10.8, 4.6, 2.9, 0.0)$
`wing_thickness_cp`	$(0.010.010.01)$
`wing_chord`	$(2.7, 3.7, 4.8, 5.8, 6.8, 8.7, 11.2, 13.6)$
`CD cruise`	0.0305

Table 7: Parameters optimized after first iteration

Some of the results obtained are tabulated in order to better compare with further iterations.

2.3 Mesh quality increased

The exact same optimization problem was performed, increasing only the quality of the wing mesh. The values for *num_y* and *num_x* where set to, respectively, 21 and 5. The results are tabulated in table 8.

Parameter	Value
`alpha`	7.46
`wing_twist_cp`	$(4.1, 8.4, 8.0, 1.1, 0.0)$
`wing_thickness_cp`	$(0.010.010.01)$
`wing_chord`	$(2.7, 3.5, 4.2, 4.9, 5.6, 6.3, 7.0, 8.5, 10.2, 11.9, 13.6)$
`fuelburn`	2074

Table 8: Parameters optimized with higher quality mesh

Through the analyses of the results it is clearly visible that, despite small changes, most of the parameters are either equal or really closed to the ones computed with a coarser mesh. It is possible to conclude that it is not necessary to have an high quality mesh in order to obtain feasible and accurate results. Also, the number o iterations and function evaluations increased significantly. The time taken to perform this optimization was around 25min, whereas the previous one took about 5min. The computational cost increases highly, and the results are not improved.

Parameter	Value
Exit mode	0
Current function value	0.0207
Iterations	100
Function evaluations	103
Gradient evaluations	100

Table 9: Optimization parameters

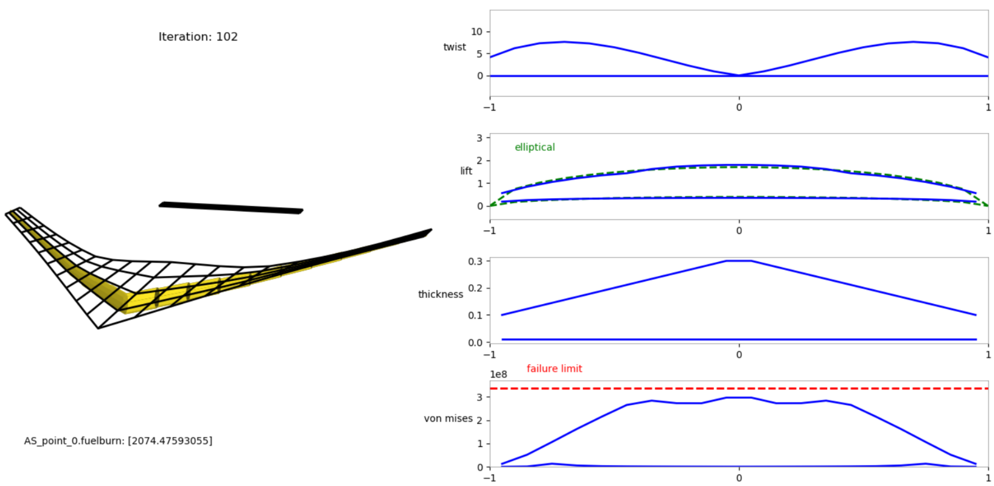

Figure 3: Results of first iteration (*plot_wing* script) for an high quality mesh

From now on, the surfaces will be divided along the span direction by 17 lines and along the chord direction 3 times. This way, optimization will occur in less time and computational cost will be reduced without interfering with the results.

2.4 `Alpha maneuver` as design variable

The next design variable added was `alpha maneuver`. This angle defines the flow direction during the maneuver and connections between variables have to be made in order to differentiate from `alpha cruise`. This modification was made in order to have a better defined condition at the maneuver point and, furthermore, have a more accurate estimate of structural stresses that could have implications on other design variables.

Parameter	**Value**
`alpha`	10
`wing_twist_cp`	$(2.2, 8.8, 2.9, -1.7, 0.0)$
`wing_thickness_cp`	$(0.01, 0.01, 0.01)$
`wing_chord`	$(2.7, 3.6, 4.5, 5.4, 6.3, 7.3, 9.3, 11.5, 13.6)$
`alpha maneuver`	1.84
`CD cruise`	0.0305
`fuelburn`	2004

Table 10: Parameters optimized with `alpha maneuver` as design variable

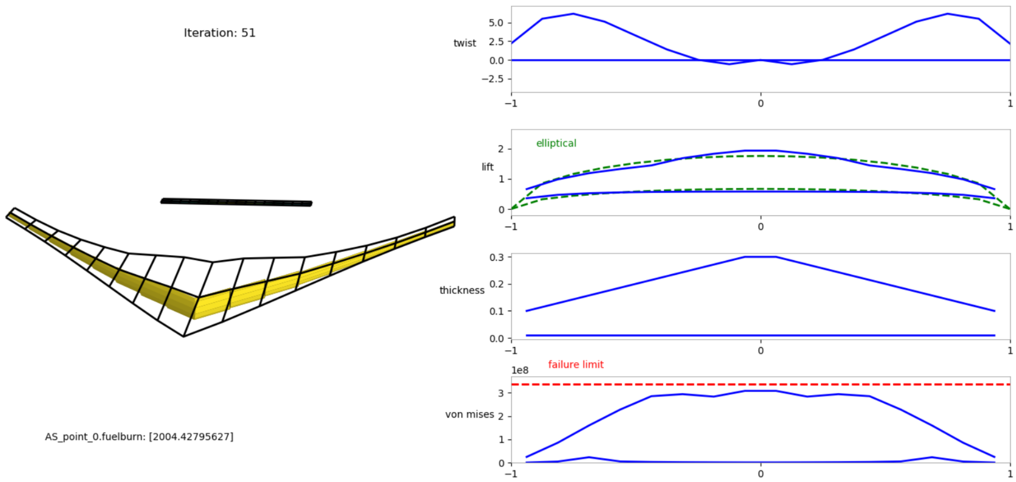

Figure 4: Results having `alpha maneuver` as design variable (*plot_wing* script)

The lift distribution remains practically constant. As for the wing twist, it now shows an oscillatory pattern, having negatives values near the chord. The stress, which is the variable expected to change the most, also remains similar. The biggest difference is visible on the `angle of attack` at cruise, which is now equal to the limit imposed.

2.5 `Wing sweep` as design variable

The next design variable added was the wing swept. Adding this variable to the problem there are now 5 design variables which increases the complexity of the optimization. It is expected for the number of iterations and function evaluations to increase, which does not happen as shown bellow. For this optimization, results are summarized in table 12.

Parameter	**Value**
`alpha`	5.45
`wing_twist_cp`	$(-0.7, 3.3, 0.6, 0.7, 0)$
`wing_thickness_cp`	$(0.01, 0.01, 0.01)$
`wing_chord`	$(2.7, 3.6, 4.5, 5.4, 6.3, 7.3, 9.3, 11.5, 13.6)$
`alpha maneuver`	0.085
`CD cruise`	0.0273
`wing sweep`	-28.9
`fuelburn`	1718

Table 11: Parameters optimized with `wing sweep` as design variable

Comparing values for `wing twist` it is verified that the maximum twist decreased significantly and that overall the values are much smaller. The value for `CD cruise` also decreased significantly, so as the objetive function, fuelburn. However, looking at the shape of the wing shown in figure 6 it now has a weird shape, different from most wings used in commercial aircrafts. Even though it might me a feasible solution, it certainly isn't the optimal one, which once again is a reminder of the limitations of a low-fidelity tool.

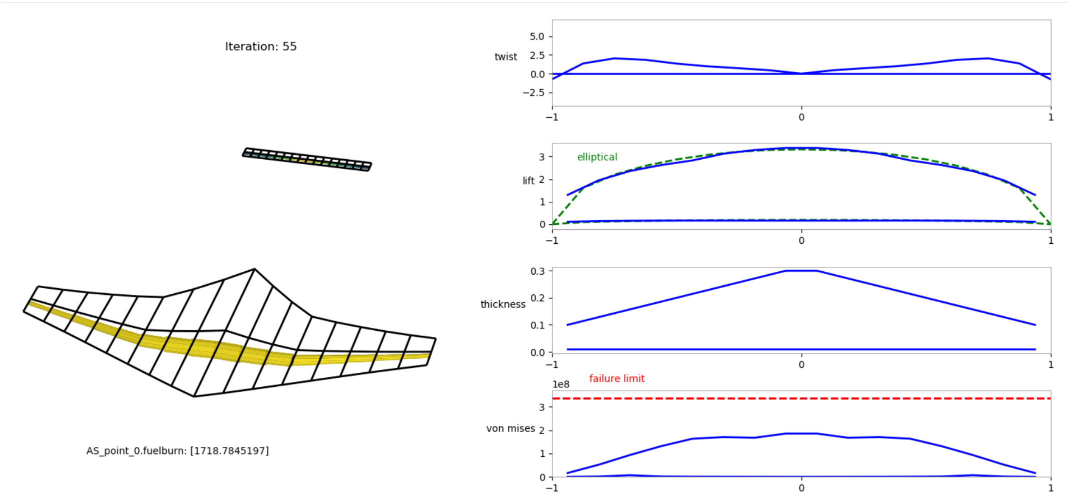

Figure 5: Results having `wing sweep` as design variable (*plot_wing* script)

In order increase the complexity of the problem and also trying to obtain results better adjusted than the previous iteration, the `wing span` was defined as a design variable. So far, the wing span was defined in reference to the Bombardier aircraft. After concluding the simulation two main aspects were concluded. Firstly, the number of iterations increased significantly to over 80, showing that the complexity of the problem starts having weight on the computational cost. Secondly, the solver was not able to define a feasible line search to optimize the wing span. The maximum value allowed was 40m and it reached this value, which is clearly an unfeasible solution to put in practice. Thus, the wing span was removed from being a design variable and was set to the original value.

2.6 `Tail twist` as design variable

For the first time, the tail will now start to be a part of the optimization problem. The `tail twist` is added to the design variables group in order to understand the impact that the tail configuration can have on the main wing and flight conditions.

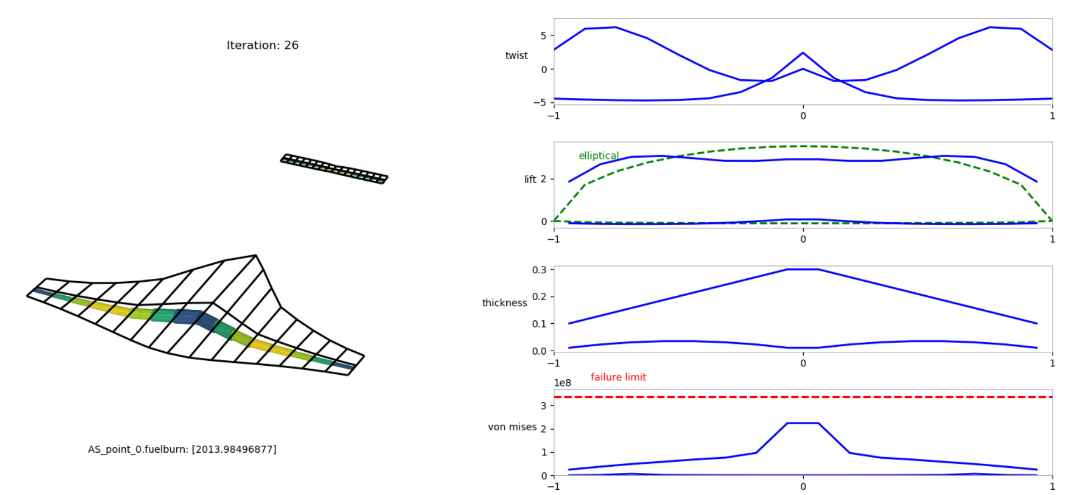

Figure 6: Results having `tail twist` as design variable (*plot_wing* script)

Parameter	**Value**
alpha	4.93
wing_twist_cp	$(2.8, 9.4, 1.5, -4.0, 0.0)$
wing_thickness_cp	$(0.01, 0.05, 0.01)$
wing_chord	$(2.7, 3.6, 4.5, 5.4, 6.3, 7.3, 9.3, 11.5, 13.6)$
alpha maneuver	-0.149
CD cruise	0.0290
wing sweep	-34.0
wing_twist_cp	$(-4.4, -4.6, -5.0, -4.0, 2.4)$
fuelburn	2013

Table 12: Parameters optimized with `wing sweep` as design variable

The influence on having the tail twist as design variable is much higher than expected. The oscillatory behaviour that was already found increased severely, which make it an option to discard. It is also important to mention that for the first time the thickness changed slightly. The wing sweep also decreased and the fuelburn increased, contrary to pretended. This time, the number of iterations decrease to only 24, suggesting that sometimes it is unpredictable to estimate the difficulty to find a solution, no matter how many design variables are being used. To conclude, while increasing the number of design variables might seem as an improvement on the final solution, it not always happens. The solutions obtained firstly, where the number of design variables was still reduced seemed a more feasible solution.

3 Optimization problem with wing-box model

The optimization problem will be mainly similar to the one presented above. However, in order to further learn about built-in tools of OpenAeroStruct, this time the FEM model will be the wing-box. This optimization will only include only the main wing due to problems converging when using both wing and tail. The first step taken was to understand how the wing-box model works and what built in function could serve as constraints. The wing-box model is commonly used since typically, transport aircraft wings are designed with structures called wing-boxes. These are composed by two skins (lower and higher) and two spars.The spars are responsible for supporting mainly the shear load, while the skins support mainly the bending loads. In order to run a simulation it is required for the user to specify the location of the wing-box coordinates. This coordinates correspond to the ones of the airfoil. The airfoil chosen was the NASA SC2-0610, which is a supercritical airfoil used in transonic flights to improve shock wave drag. The airfoil coordinates were obtained on the Airfoil Tool library. There are other variables that play a role on this optimization such as moments of inertia, torsion constant and thickness of skin and spars. The torsion constant is calculated assuming constant shear flow across the wall thickness for a closed section. The situation that is expected to subject the wing to higher stress is not cruise flight condition. For this reason, the optimization will have to distinct analyses points. The first one will be a cruise level flight at the specified altitude. The second will be a 2.5g maneuver at sea level conditions. Once again, this will allow the analyses of stress at the maneuver point, constraining the parameters and ensuring that the aircraft will not only resist the stresses carried at cruise but also in a maneuver. This will improve the structural performance since the shear forces and bending moments are higher during the maneuver. The optimizer used will be ScipyOptimizeDriver default, which is the 'SLSQP'. Some of the default values for this method are shown in table 1.

$$\begin{cases} minimize \quad \texttt{fuelburn} \\ w.r.t. \quad \texttt{wing twist}, \texttt{thickness-to-chord ratio}, \texttt{spar thickness} \\ \qquad \texttt{skin thickness}, \texttt{maneuver angle of attack} \\ subjet \quad to \qquad C_{Lcruise} > 0.45, L_{maneuver} = W_{maneuver}, \sigma_{VonMises} \geq \sigma_{lim} \\ \qquad \texttt{fuel volume} \leq \texttt{wingbox volume}, \texttt{twist at root} = 0 \end{cases} \tag{4}$$

Parameters	**Default**
Algorithm	'quasi-SLSQP'
MaxIterations	200
Tolerance	1×10^{-6}

Table 13: Default values for ScipyOptimizeDriver

Before presenting some results, it is important to take a look at some of the values used to defined some parameters that helped to get the first optimization results.

Variable	Value	Units
num_x	17	-
num_y	3	-
num_twist_cp	6	-
span	23.24	m/s
taper_ratio	0.3	
C_{L_0}	0.1	
C_{D_0}	0.0078	
$\frac{c}{t}_max$	12%	
chord for $\frac{c}{t}_max$	0.37	
Young's modulus	71.7×10^9	Pa
Shear modulus	26.9×10^9	Pa
Maximum yield stress	$503 \times 10^6/1.5$	MPa
Density	2810	kg/m^3
wing_weight_ratio	1.25	
Fuel density	803	kg/m^3

Table 14: Aerostructural parameters used

The structural values were obtained through some research on the alloy Al 7075-T6. The parameter num_twist_cp represents the number of spline points that will be used to define the twist of the wing. As previously mentioned, the optimization will be mainly constituted by two flight condition points that are defined in the previous section on tables 4 and 2.

3.1 N2 diagram

The N2 diagram will once again help the user to have a better understanding of how the variables are being transferred from one module to the other. In this way, the following figure is shown in order to comprehend the dynamics of the optimization problem.

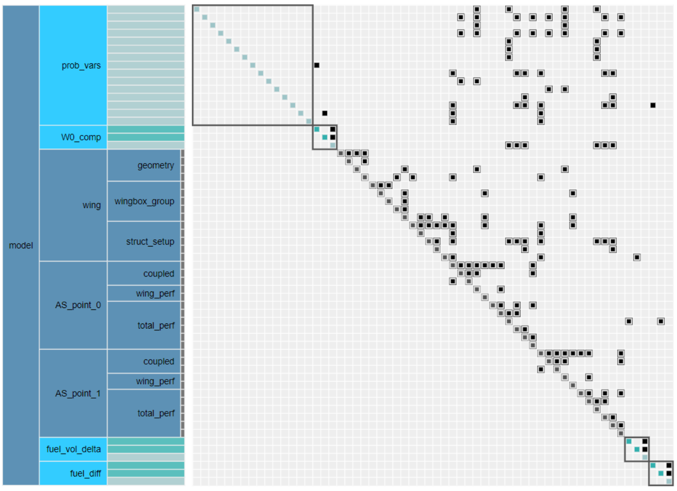

Figure 7: N2 diagram for the wing-box model problem

In figure 7 the major groups are easily identified. The *wing* group contains all the geometric information that defines the lifting surface considered. It is in this group that the calculations related to the sectional moments, moments of inertia take place. This calculations are what differ from using the *tube* model. Moving on to the *AS_point_0* group, it contains all the calculations performed to obtain forces such as lift and drag in the cruise flight point. Finally, the *AS_point_1* corresponds to the maneuver point. It is very similar to the cruise condition group since it is also responsible for the calculation of forces, stresses and so on. This model and corresponding components are connected as seen by the black squares. Once again, it is really important to pay attention to all the variables to check for unconnected ones. If, for instance, one independent variable is not connected to all the components where that variable is needed, the program will either crash or assume a default value, which may be harmful for the results.

3.2 Results

Parameter	**Value**
Exit mode	0
Current function value	0.0321
Iterations	69
Function evaluations	72
Gradient evaluations	69

Table 15: Optimization parameters

This simulation is successful, since the exit mode is 0. Regarding the number of iterations and function evaluations, the number is quite similar for the ones obtained for the previous simulation. Baring in mind that there are 5 design variables, one can say that these numbers are actually quite low.

With the aid of the script *plot_wingbox* it is possible to obtain the plot of several variables and have a graphical representation of the wing and corresponding wing-box. This tools plots the variables that we are interested in, such as the normalized lift, twist, thickness to chord ratio, thickness of the skin and spar and the Von Mises stresses along the wing span.

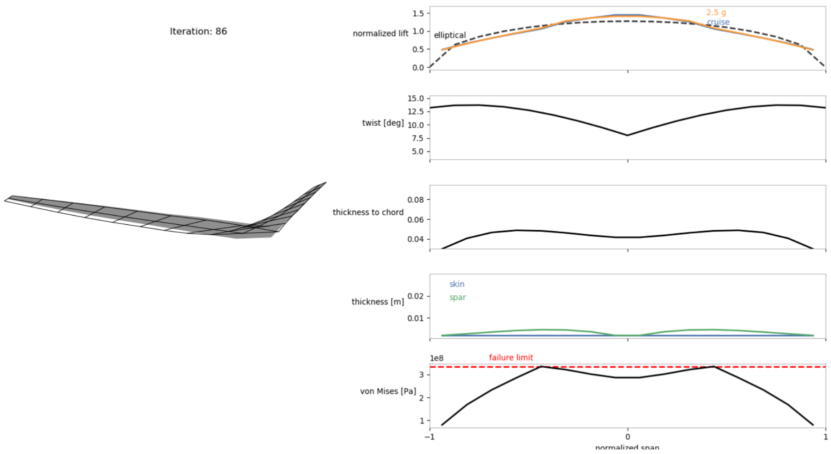

Figure 8: Graphical representation of the wing

Getting into the analysis of figure 8 it is possible to conclude that the generated lift follows the perfect solution (elliptical lift distribution), having higher values at the root section, and lower values when increasing the distance to the wing root. As for the twist, the smaller value is as expected in the wing root and increases along the wing span. However, the value for the twist at the wing root is not constrained and thus has a high value, far from 0 degrees. The thickness to chord ratio presents a the lowest value at the wing tip. For the first time, the structural analyses have a big influence in this optimization. One can easily conclude that the failure limit is reached, constrain some of the design variables, mainly the spar and skin thickness. In this optimization it is being used a built in function as a constrain, which guarantees that the volume inside the wing-box is high enough to store the fuel needed for the specified range. This is highly important in practice since most air crafts store fuel in the wings. When comparing the value obtained for the fuel mass with values from the previous section it is possible to notice a discrepancy. Using the wing-box model this value is more than 60% higher. This suggests that one of the values might be incorrect, due to some user specified input or due to the way it is calculated depending on the model. The wing structural mass is about 4% of the take off weight which is a good estimate.

Variable	Value
Wing twist @cp	$(12.7, 15.0, 11.7, 7.9)$
Wing twist	$(12.7, 13.3, 13.5, 13.2, 12.6, 11.7, 10.6, 9.3, 7.9)$
Spar thickness	$(0.0020, 0.0038, 0.0072, 0.0042)$
Skin thickness	$(0.0020, 0.0020, 0.0020, 0.0020)$
$\frac{t}{c}_{cp}$	$(0.020, 0.071, 0.043, 0.042)$
$\alpha_{maneuver}$	-4.60

Table 16: Design variables results

Variable	Value
C_L	0.474
C_D	0.028
Fuel mass	3212.0
Wing structural mass	1318.5
$\alpha_{maneuver}$	-4.60

Table 17: Other parameter results

4 Conclusion

After the realization of all the mentioned simulations it is plausible to say that OpenAeroStruct has has many short falls while solving design problems. These problems are mainly the difficulty to achieve a plausible solution that is ready to implement or convergence problems. However, baring in mind that OpenAeroStruct is a low-fidelity software, good estimates for the first stage of analyses can be computed and improved in an higher fidelity software at later stages of the design process. The main issued encountered was that frequently, the software was not able to converge, and, being the user, a lot of time was wasted while waiting for a solution that would never converge. Another struggle is that when the software does not converge, the reason is not specified and it is really hard for the user to understand the reason. When adding constrains, for instance, it was really hard to find constrains that would improve the solution and, at the same time, allow the problem to converge. Overall, it is a fact that OpenAeroStruct serves perfectly at early stages of a design to have an initial solution, mostly because it is a very lightweight program and also free. Once the user is trained to understand all the details behind it, it is a very powerful tool.

References

[1] *OpenAeroStructDocumentation* https://mdolab.github.io/OpenAeroStruct/more_examples.html

[2] *U.S.StandardAtmosphere* https://www.engineeringtoolbox.com/standard-atmosphere-d_604.html

[3] *Aluminum 7075-T6* http://asm.matweb.com/search/SpecificMaterial.asp?bassnum=MA7075T6

[4] *Bombardier CRJ700* https://en.wikipedia.org/wiki/Bombardier_CRJ700_series