



WING AERODYNAMIC DESIGN FOR A MAME UAV USING HIGH-FIDELITY NUMERICAL TOOLS

Rúben S. Gameiro¹, Nuno M. B. Matos² and André C. Marta¹

1: IDMEC
Instituto Superior Técnico
Universidade de Lisboa
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
{rubengameiro, andre.marta}@tecnico.ulisboa.pt, <https://mdo.tecnico.ulisboa.pt>

2: Research and Development
Tekever UAS
Tekever
2500-750 Caldas da Rainha
nuno.matos@tekever.com, <https://tekever.com>

Abstract. *The UAV market is currently very competitive, with the frequent launch of new products and a wide range of solutions already available, forcing manufacturers to explore the design space faster and more efficiently than in the past. A cost effective approach is to develop growth versions, improving an existing product with new technologies and design tools. Some of these tools include RANS based high fidelity computational fluid dynamics methods and discrete adjoint gradient-based optimization, which will be used in this work on a numerical design framework to explore the aerodynamic shape optimization of a wing, as part of the development of a growth version of a MAME UAV for a leading Portuguese manufacturer. A comprehensive aerodynamic analysis of the current UAV wing will be performed, followed by an optimization procedure to minimize drag subject to a prescribed lift coefficient constraint. To that end, two different starting geometries will be considered and parameterized with common design variables, including twist and chord distribution, sweep and airfoil shape. New optimized geometries for different sets of design variables will be obtained with a significant drag coefficient reduction from the starting geometry. The optimized geometries will approach an elliptical lift distribution, although not exactly considering the trade-offs needed between skin friction and induced drag. Despite the fact that the results obtained here are not considered the final design, as more shape parametrizations and design variables are yet to be explored, they provided a good insight on how the different parametrizations are handled by the design optimization framework and considerations that should be taken.*

Keywords: Optimization, adjoint method, aircraft design, computational fluid dynamics, free-form deformation, geometric parametrization

1 INTRODUCTION

UAVs have been around for more than a century [1]. Initially, developments went hand in hand with the recent aviation industry, following a technology-driven approach which allowed a fast evolution, benefiting from the emerging technologies in all disciplines (aerodynamics, structures, flight mechanics, etc.) [2].

As speeds reached by manned aircraft started to increase significantly during the 60's and the 70's, more focus started to be given to aerodynamic design, starting with airfoil design and quickly evolving to wing design [3]. At this time, CFD was used primarily to obtain a better mission performance by increasing range and speed. However, with the oil crisis in the 70's, this old approach to aircraft design started to shift into a customer driven approach which is still maintained nowadays, meaning that aircraft design started to be a balance between available technology and the benefits it would bring to the customer, taking into account the required constraints [4]. Satisfying the customer needs also implied a reduction in development time so that the manufacturer could respond in time to the current market needs.

During this shift, CFD underwent very significant advances, with the solution of the viscous flow around a 3D geometry of the whole aircraft using RANS equations and turbulent models eventually becoming standard [3]. With the evolution in computational power and code scalability, it became feasible to use such models on the conceptual design phase [5–7]. CFD brought several advantages to the design process, making it much more efficient. However the project of a new vehicle can still take years, with several iterations between the different disciplines, such as structures, aerodynamics and propulsion, often on an intuition and trial and error basis to assess different possible vehicle shapes [3].

Nowadays, the UAV market is experiencing fast growth and dynamic developments, which naturally attracts investors attention. Notably, the industry received a total investment of *US*\$6.96 billion last year [8]. This growing market has also become highly competitive, with the influx of new market players resulting from the substantial investments, companies seeking external funding must differentiate themselves by offering innovative and unique products.



Figure 1: Tekever AR5 UAV. Source: Tekever.

In this context, this work explores automatic aerodynamic shape optimization using high fidelity methods, in particular the MACH-Aero framework, developed by the MDOLab [9]. The focus is on the development of a growth version of the Tekever AR5 UAV, shown in Figure 1. Several different parametrizations will be tested and associated with common design variables which, together with a prescribed lift coefficient constraint

and a drag coefficient reduction as an objective, will form the optimization problem in study. For that, the aerodynamic analysis tools associated with the framework are used first to perform an analysis of the current AR5 wing. After that, optimization problems with different sets of design variables will be set up and the results analyzed and commented.

In the future, this project will evolve to a multidisciplinary optimization (MDO) framework, where the aerodynamic and structural optimization will be performed together.

2 FRAMEWORK OVERVIEW

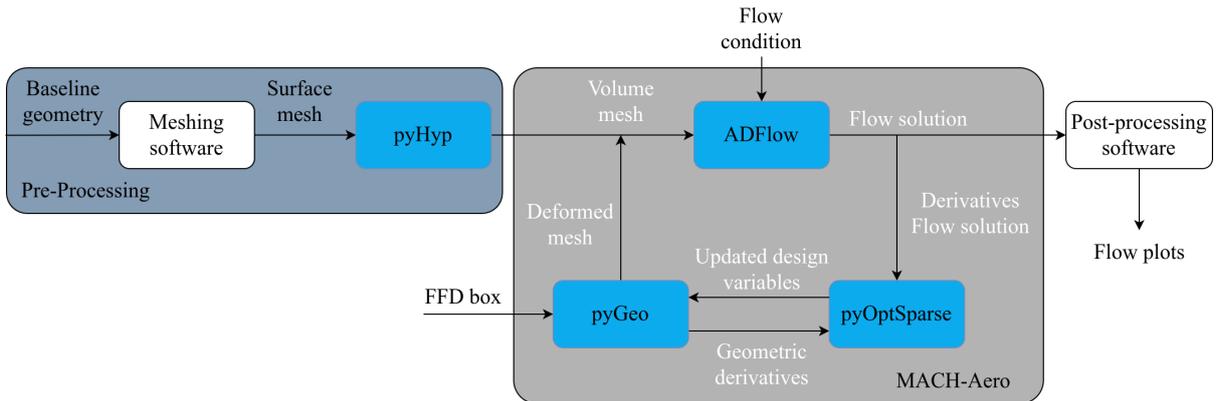


Figure 2: MACH-Aero optimization framework.

Figure 2 schematizes the process followed to optimize the aerodynamic shape of a wing. The blue shaded boxes represent the modules developed by the MDOLab and the gray box encompasses the MACH-Aero framework. The white text represents parameters that are automatically updated by the MACH-Aero framework, and the black text represents parameters that require user input.

The MACH-Framework is composed by the following modules: *ADFlow*, a structured multi-block 3D CFD solver [10] which also solves the adjoint method to compute the derivatives [11], *pyGeo* a geometry manipulation tool that was specifically built for multidisciplinary optimization applications and allows the manipulation, parametrization and constraint handling of the geometric shape [12] and *pyOptSparse*, which handles the constrained nonlinear optimization problem [13].

2.1 Meshing

The first step is to create a baseline geometry which will serve as a starting point for the optimizer. Considering that this geometry will be an input to the MACH-Aero framework, two main options are available for generating the geometry: Using *pyGeo*, that allows to create simple geometries by lofting surfaces between a provided set of airfoils in a code-based way, or using a typical CAD (Computer Aided Design) or geometry generation software, like OpenVSP [14], which was the main tool used in this work to obtain the starting point for the optimizer. The geometry should then be exported to a neutral CAD format that is accepted by the mesh generator. IGES (International Graphics Exchange Standards) or STEP (Standard for the Exchange of Product Data) are two formats widely accepted by most meshing software.

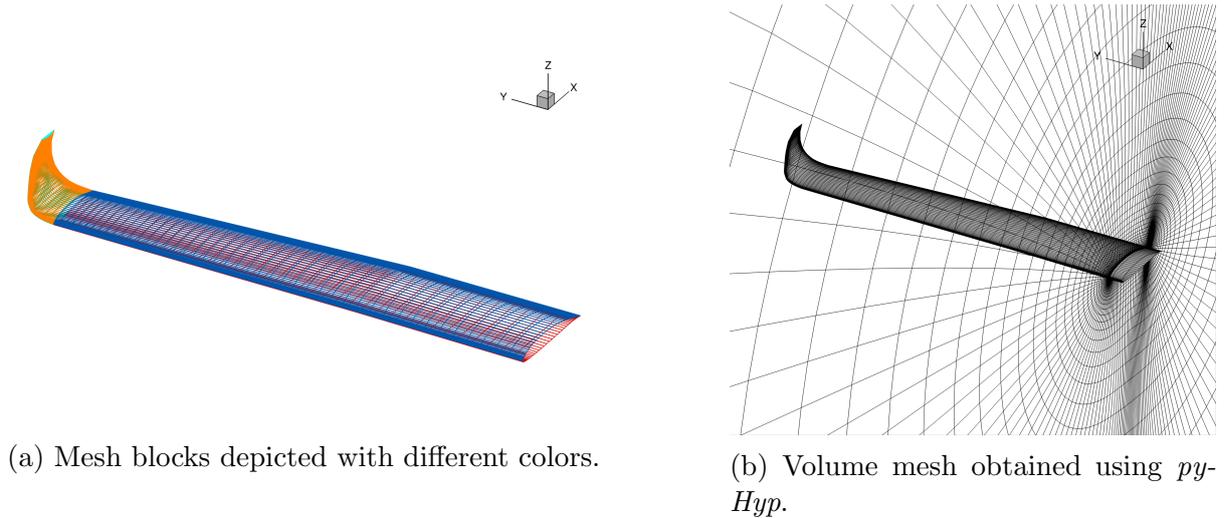


Figure 3: Used mesh for the Tekever AR5 wing.

From the geometry CAD file, a surface mesh must be obtained. For that, it is possible to choose from a variety of different software. However, the chosen grid generator should be able to generate a multi-block and/or overset 2D mesh, as those are the grid types accepted by the flow solver, and output the resulting grid to a CGNS (CFD General Notation System) file, in ADF [15] or HDF5 format, supported in more recent versions of CGNS [16].

The most popular options are ICEM CFD[®] and Pointwise[®]. Some open source solutions for mesh generation, such as GMSH [17] are also widely available. However, the first requirement limits their usage in this work. Figure 3a shows an example of a surface mesh with multiple blocks used in this work.

Having a surface mesh in the appropriate format, an hyperbolic volume mesh is then extruded using *pyHyp*, an hyperbolic mesh generator that also automatically applies the necessary boundary conditions for a wing if desired [18]. Figure 3b shows a volume mesh obtained with *pyGeo*, as well as the respective surface mesh. To generate a volume mesh from a surface mesh, *pyHyp* requires the user to provide a few parameters: the first layer height, the total height of the volume mesh and the number of layers to extrude. Both the surface and the volume mesh should, naturally, be sufficiently refined, and so, grid independence studies must be carried out before starting the aerodynamic analysis and optimization procedure.

2.2 Flow model

To compute the flow, *ADFlow* includes both inviscid models (Euler) and viscous models, which solve the RANS equations with different turbulence models available, including Spalart-Allmaras, Wilcox $k-\omega$, $k-\tau$, Menter SST $k-\omega$ and $v2-f$. In this work, it is intended to optimize the wing for viscous flow, so RANS models will be used.

Regarding the turbulence model, comparisons for external flows around an Onera M6 wing concluded that the Spalart-Allmaras and SST $k-\omega$ turbulence models gave the closest results to the experimental data [19] and using the DPW7 CRM configuration shown that both the $k-\omega$ and Spalart-Allmaras models yield great results [20]. The main drawback of the Spalart-Allmaras model is for separated flows, performing well for external flows

otherwise [21].

The flow solution using *ADFlow* with Spalart-Allmaras turbulence model was also compared with two other solvers and to experimental data for the CRM wing, and successful validation of the *ADFlow* code was achieved [22].

It is important to note that only the Spalart-Allmaras turbulence model has been differentiated in the *ADFlow* code. This fact, together with previous observations of the good results obtained with Spalart-Allmaras for external flow computations makes it the most suited for the optimization of a wing. Spalart-Allmaras is a linear eddy viscosity model, and thus, uses Boussinesq assumption for the constitutive relation [23]. Furthermore, it is considered a low Reynolds number, and in [24] it was concluded that an $y+$ value of approximately 1 was enough to obtain numerical uncertainties of friction resistance coefficients smaller than 1%.

The flow conditions are defined and passed to *ADFlow* by instantiating the *AeroProblem* class from *baseclasses*. Multiple formulations could be used, but for this work, the flow condition will be defined using the angle of attack, α , flight altitude, h , and the Mach number, Ma of the undisturbed flow. From these parameters the complete flow state information is obtained from the 1976 U.S. Standard Atmosphere [25]. Furthermore, wing reference area and chord should be passed as inputs to the *AeroProblem* for the computation of the aerodynamic coefficients.

2.3 Geometric parameterization

Using the MACH-Aero framework, there are two main options for geometry parametrization: CAD-based and free-form deformation based, which uses a box that completely embeds the surface mesh and that will be referred in this work simply as the FFD box. The surface mesh nodes are then mapped to the FFD box with an $\mathcal{R}^3 \rightarrow \mathcal{R}^3$ mapping, determined by performing a Newton search. After that, surface mesh nodes can be deformed by performing deformations on the FFD box nodes, given their mapping. The surface mesh deformations are then used to perturb the volume mesh using an hybrid algebraic-linear-elasticity mesh perturbation scheme [12] due to its resulting high grid quality and low computational effort. Figure 4 shows the result of using such scheme to deform the surface mesh of a wing, where it can be seen that the deformation of the FFD box deformed the embedded wing.

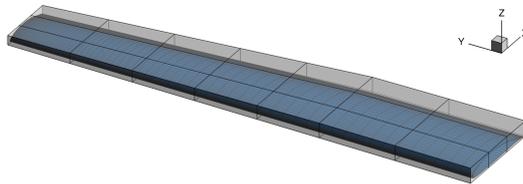


Figure 4: FFD box used to deform a wing surface mesh.

CAD-based approaches parameterize the geometry directly and so the design variables for the optimizer are directly related to parametric CAD variables, which also makes it easier to use the optimized geometry. However, the FFD approach is generally easier to setup and use with an already existing geometry and has freedom to parametrize multiple design variables easily, so it was the chosen approach.

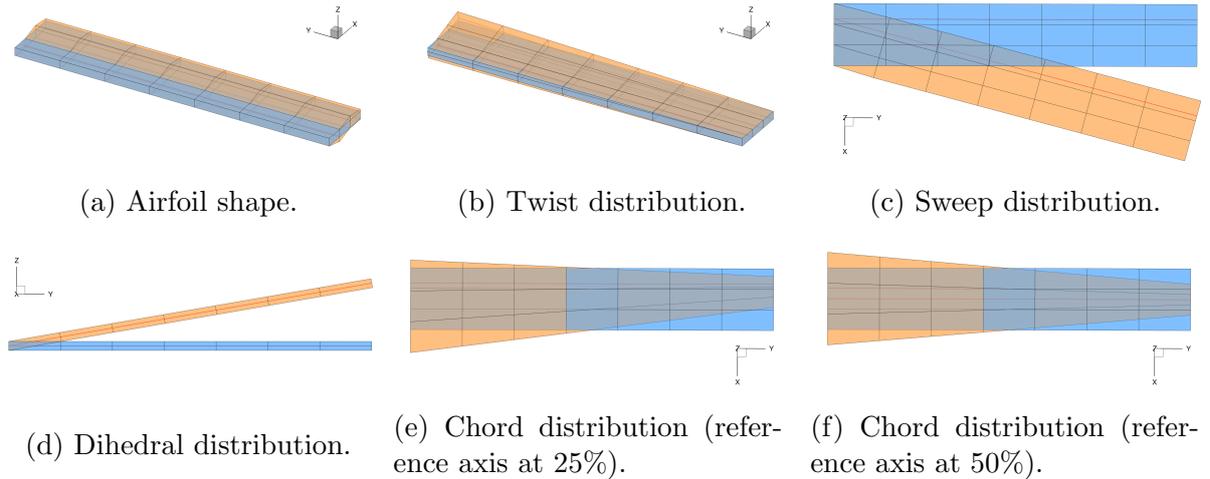


Figure 5: Deformations to the FFD box associated with common design variables.

The nodes of the FFD box may be displaced individually in any spatial direction, and will, in that case, be known as local design variables. However, *pyGeo* also allows to create relations between node displacements in such a way that they are all affected by a smaller set of design variables, allowing the geometry parameterization with global design variables. This second approach is useful as the reduced number of design variables saves computational resources during the optimization whilst allowing for the creation of functions that can represent more intuitive deformations such as taper or twist. An example of a local design variable with vertical displacements is given in Figure 5a where the parameterization was set up such that all sections of the FFD box would mimic the same deformations.

To create a global design variable using *pyGeo*, in addition to the FFD box, an axis is necessary. The axis can be created by specifying its direction and relative position in the FFD box using the fraction of two specified directions. For a typical wing optimization problem, the user should specify the direction that follows the wingspan, and the fraction of the vertical and streamwise position of the axis, which will be, for most practical cases, 0.5 and 0.25 respectively, representing the quarter chord of the wing. It is important to note that the wing should be centered within the FFD box. The reference axis will be used to project all the FFD nodes into it using an user defined direction for the projection. The points will then become rigidly linked to the reference axis, and so, any deformation on the FFD axis will have an effect on several FFD points at once [26]. A deformation on the FFD axis may be of three types: displacement, rotation and scaling.

The first type of deformation is the displacement of the points to a new location, which may be used to create dihedral or sweep, for example, as Figures 5d and 5c show. *pyGeo* also allows rotations and scaling in any one of the three directions x , y and z . The most obvious application of a reference axis rotation would be to create a twist distribution, depicted in Figure 5b, but it is also useful to keep the airfoil sections perpendicular to the reference axis and, thus, effectively keeping the same airfoil by applying rotations to the reference axis points across the axis of rotation. This effect can be clearly visible in Figures 5d and 5c. The scaling function can scale the FFD box points in a specified direction. Figures 5e and 5f show an example of scaling functions being applied to create a chord distribution. Besides the obvious scaling in the streamwise direction, a scaling

in the vertical direction was also applied to maintain the airfoil thickness to chord ratio constant. While Figure 5e shows a chord distribution for a reference axis at 25% of the FFD box, Figure 5f shows a chord distribution for a reference axis at 50%.

The design variables are not necessarily defined by a single number and can have as many degrees of freedom as the number of points in the reference axis. It is even possible to parametrize distributions of the variables with smaller sets of variables: A linear distribution of the chord or twist can be parametrized with only the slope as the design variable.

2.4 Constraints

pyGeo can also be used to set geometric constraints. It is possible to add different types of constraints, being the most common ones the minimum thickness constraints, to ensure that there is enough room for structural components; the minimum volume constraints, to guarantee enough internal volume to carry a specified amount of fuel; the curvature constraints, typically used to ensure manufacturability; and LeTe constraints to avoid shearing twist at the leading and trailing edge when local design variables are present.

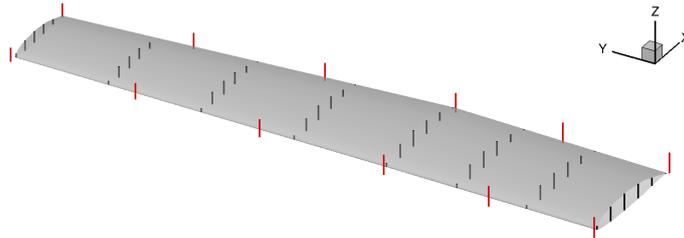


Figure 6: Minimum thickness and LeTe constraints defined using *pyGeo*.

Minimum thickness and volume constraints may be specified with either an absolute value or a relative value in relation to the baseline design. Figure 6 shows an example of the used minimum thickness (represented in black) and LeTe (shown in red) constraints. They are only enforced at certain user-specified locations and are independent of the FFD nodes. When the geometry is deformed, the change of location of those constraints is also handled automatically by *pyGeo*, which also takes care of the needed operations to use them as constraints in the optimization problem.

In this work, only minimum thickness constraints and LeTe constraints will be used. Minimum volume constraints are not required, considering that the UAV in study does not store the fuel in its wings.

2.5 Optimization problem

With both constraints and design variables defined in *pyGeo*, they need to be passed to *pyOptSparse* for the optimization process. Variables from the aero problem, defined using the *baseClasses* can also be passed as optimization variables and a particularly important example of that is the angle of attack. When adding the angle of attack as a design variable, attention must be paid to its relation with twist, given that a geometric rotation of the root wing section has an equivalent effect to the angle of attack. For this

reason, twist at the root is usually kept fixed and not added as a design variable. This also ensures that the shape of the wing mounted at the fuselage is conserved, and thus, the same principle may be applied to the other design variables.

Non-geometric constraints can also be created by user defined functions. The most important non-geometric constraint when optimizing for cruise flight is the lift constraint, which should equal a prescribed lift coefficient at cruise condition and ensure that it is possible to sustain flight. In a similar way, it is possible to define the function or combination of functions to be used as the optimization objective. In this case, the optimization objective will be the drag coefficient minimization.

With the design variables, usually defined by *pyGeo* parametrizations, constraints from both the geometry and the AeroProblem and the objective function passed to the optimization problem, the optimization set-up is effectively complete and ready to run. Given that *pyOptSparse* sets up the optimization in a problem oriented manner, a problem statement in the standard form is easily translated to the optimization framework.

When running, the CFD solver will compute the solution for the baseline geometry and the adjoints, which will then be fed back to the optimizer. The optimizer will use this information to update the value of every design variable and fed them back to *pyGeo*, which will deform the volume mesh as described before. The new volume mesh will be solved by *ADFlow* again, and when performing multiple function evaluations in the same optimization iteration, the solver will take the final solution of the last call as a starting point, speeding up iterations. This process will be repeated until convergence is achieved.

pyOptSparse can use different optimizers, including both gradient based and gradient-free ones. Both gradient-based and gradient-free optimizers have been used in the past for aerodynamic shape optimization. Comparative studies of both methods were performed in [27] and [28], with similar conclusions between them: Both can converge to a solution, but gradient-free methods are considerably more computationally expensive, which the difference increasing dramatically as the number of design variables increases. Examples of gradient-based have been performed where both refinement and exploratory studies were done using both inviscid and viscous models for the optimization of a wing using the MACH-Framework and the SNOPT optimizer and where a more complex problem of multi-component aerodynamic optimization for a wing propeller coupling was also solved using gradient-based methods within the OpenMDAO/MPhys framework [29, 30].

Although in this work a smaller set of design variables will be used initially, the number will be increased later stages, and so, the gradient-based methods were the best option. Furthermore, the functions to be used are smooth and C_1 -continuous, as the previously presented papers have shown.

The SLSQP (Sequential Least Squares Quadratic Programming) was used due to its open source nature and robustness. It has been used in the past for aerodynamic shape optimization [31, 32]. The SLSQP algorithm solves constrained nonlinear optimization problems, which corresponds to the nature of the problem being studied in this work, using the Han-Powell quasi-Newton method with BFGS update of the B-matrix and an L1-test function in the step length algorithm [33].

To compute the gradients needed for the optimization algorithm, given that in aerodynamic shape optimization problems the outputs are often reduced to the drag coefficient but multiple design variables are used as inputs, the adjoint method is the most suitable option for aerodynamic shape optimization and is the one used in the MACH-Aero

framework.

3 INITIAL GEOMETRY ANALYSIS

3.1 Operating conditions

As stated in Section 1, the Tekever AR5 wing is to be optimized. For this reason, the first step is to assess the performance of the current. Its cruise conditions are summarized in Table 1.

Table 1: Performance parameters of Tekever AR5 [34].

Cruise speed	U_∞	100km/h
Cruise altitude	h	1000ft
Maximum Take-off Weight	$MTOW$	180kg
Endurance	E	12h

The analysis should be performed for a specific angle of attack, which can be determined by optimization with a single design variable and a prescribed lift coefficient as constraint or with a secant method provided by *ADFlow* to find the initial angle of attack for a certain lift coefficient.

Tekever provided the wing lift coefficient $C_{L_{wing}} = 0.8932$, considering the projected area of the AR5 wing, $S_{wing} = 2.1691m^2$, and the cruise conditions presented in Table 1.

Despite being automatically calculated by the *AeroProblem* class, it is important to estimate the Reynolds number to understand what kind of simulation will be performed and choose the models adequately. Reynolds number is given by

$$Re = \frac{\rho U_\infty c_{MAC}}{\mu}, \quad (1)$$

where c_{MAC} is the mean aerodynamic chord, U_∞ is the free-stream velocity, defined by the cruise speed (Table 1) and air density, ρ and dynamic viscosity, μ , which can be computed for the cruise altitude using the ISA Standard atmosphere. Using all the described values, it is possible to obtain a Reynolds number of $1.1e6$. This is a moderate Reynolds number, and it indicates that turbulent flow will be dominant. The effects of viscous forces will be much smaller than the ones caused by pressure forces, but they should not be ignored.

Mach number is another dimensionless parameter that should be calculated, not only because it provides an insight into the kind of flow to be expected, but also because it is an input for the *AeroProblem*. It is given by

$$M = \frac{U_\infty}{\sqrt{\gamma RT}}, \quad (2)$$

where γ is the adiabatic constant, R is the gas constant and T is the temperature, calculated also with the ISA Standard atmosphere for the cruise altitude, and with a value of $286.169K$. This results in a Mach number of 0.08164 , which is very low, so no transonic effects are to be expected.

Table 2 summarizes the main parameters used for the CFD simulation. Other parameters were kept at their default values [35].

Table 2: CFD solver parameters used for the simulation.

Discretization	Central plus scalar dissipation
Equation type	RANS
Equation mode	Steady
Turbulence model	Spalart-Allmaras
Turbulence order	First order

For the stopping criteria, either a maximum of 10000 iterations or a convergence of the residuals L2-norm to $1e-6$ in relation to the residuals obtained in the first iteration will be used.

3.2 Geometry

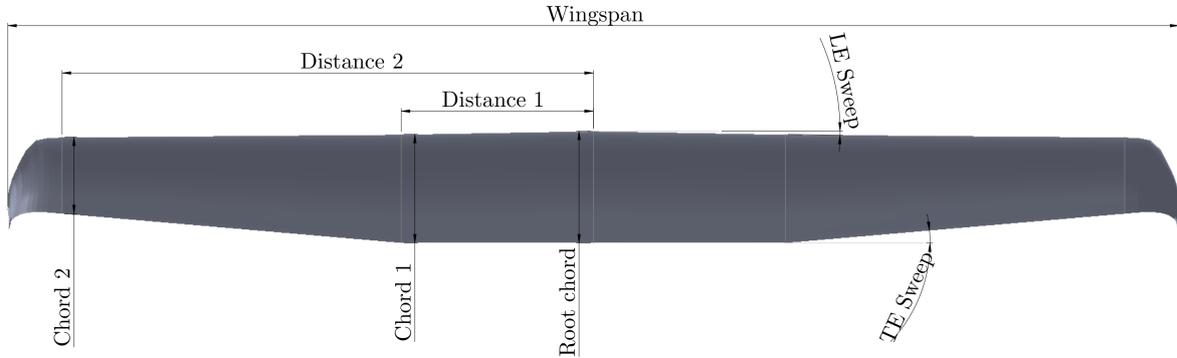


Figure 7: Planform of the AR5 wing.

The planform of the current AR5 wing is shown in Figure 7 and the main design variables associated are presented in Table 3. From distance 2 onward, the wing is composed of a winglet which will not be described here, as it will not be considered in the optimization phase of this work, but it effectively increases the real wingspan whilst keeping the projected wingspan constant and reduces the induced drag, as it will also be shown later. Table 3 also includes the surface area, as well as the projected area of this geometry, which will be kept constant for the initial geometries generated for the optimization. A custom airfoil is used.

Table 3: Some geometric parameters of Tekever AR5 wing.

Wingspan	b	7.565 m
Projected wingspan	b_{proj}	7.246 m
Root chord	c_{root}	0.70 m
Projected area	S_{proj}	2.169 m ²

3.3 Grid convergence study

A grid convergence study was performed, with an approximate uniform refinement ratio of 1.15 between grids by simulating the Tekever AR5 wing at an angle of attack of 1.5° . Sufficient grid refinement will also be important for optimization, namely chord

distribution optimization, during which the mesh in the streamwise direction will get compressed or enlarged together with the geometry, effectively changing its refinement level. This effect will be particularly relevant at the wingtip, where induced drag is expected to be captured.

For those reasons, the used grids all have a greater element clustering near the leading and trailing edge in the streamwise direction, as well as near the tip in the spanwise direction. Since no wall functions were used, the Spalart-Allmaras turbulence model requires an $y^+ \approx 1$, which was satisfied.

Figure 8 shows the evolution of the drag coefficient, as well as the required computational wall time for the different meshes tested. It is possible to observe that the drag coefficient is converging as the grids are refined. The increase in the wall time needed is also noticeable, with the more refined grid taking more than three hours to converge running on two cores of a CPU with a clock speed of $4.5GHz$. A linear increase in the needed RAM memory is verified, at around $2GB$ per $100,000$ elements. Based on this data, it can be concluded that the mesh with about 1.1 million elements is sufficiently refined, as the relative drag coefficient error to the previous iteration is only 0.36% and the computational time needed is less than half of the more refined grid.

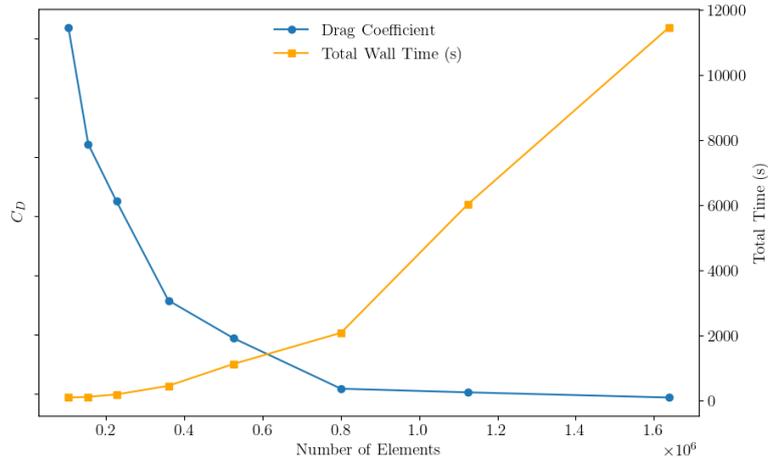


Figure 8: Grid convergence study.

3.4 Domain size study

The influence of the domain size in the solution was also evaluated. For this, several grids with different extrusion distances from the surface normal were tested. However, a coarser surface mesh was used in order to save computational resources, which will increase as the extrusion distance increases. The volume mesh was extruded from the surface mesh using *pyHyp*, which uses a geometric progression to calculate the marching distance of each new layer of cells, given by

$$\Delta d_1(1 - q^{n_k - 1}) - d(1 - q) = 0, \quad (3)$$

in which Δd_1 corresponds to the marching distance of the first layer from the surface, q is the geometric progression ratio, n_k is the number of elements in the off-wall direction

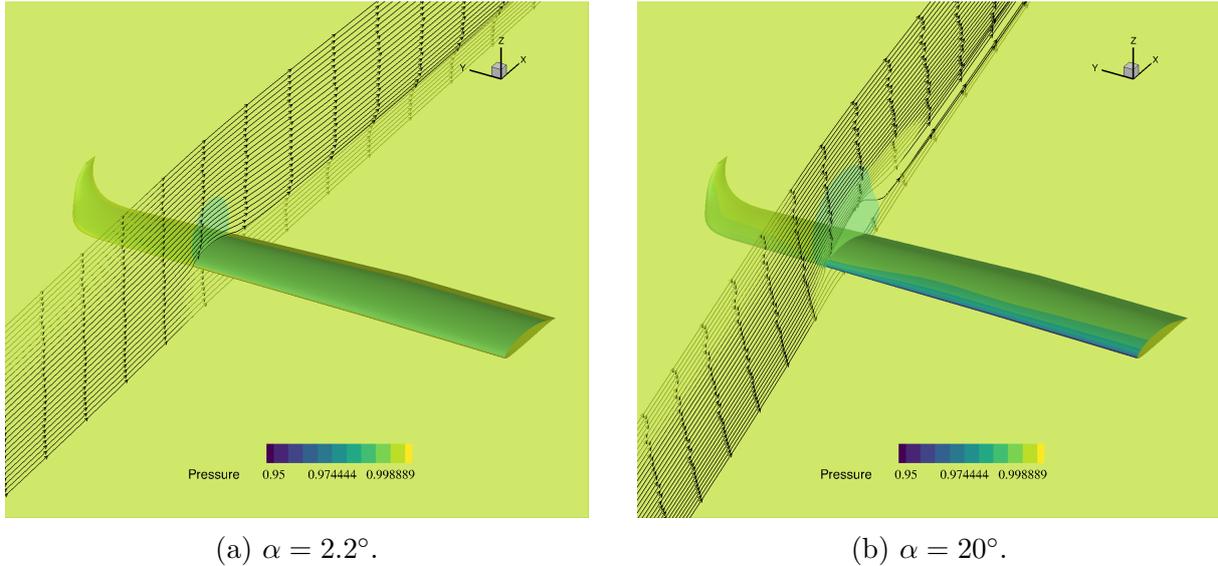


Figure 9: Pressure distribution and streamlines for different angles of attack.

and d is the total marching distance. Here, the influence of d is studied. However, the geometric progression ratio needs to be kept constant to ensure that the results are not affected by the mesh refinement in the off-wall direction, so n_k is also be changed, according to Equation 3 and q will be kept at around 1.187. The domains are measured in terms of mean aerodynamic chords, and the results are presented in Table 4.

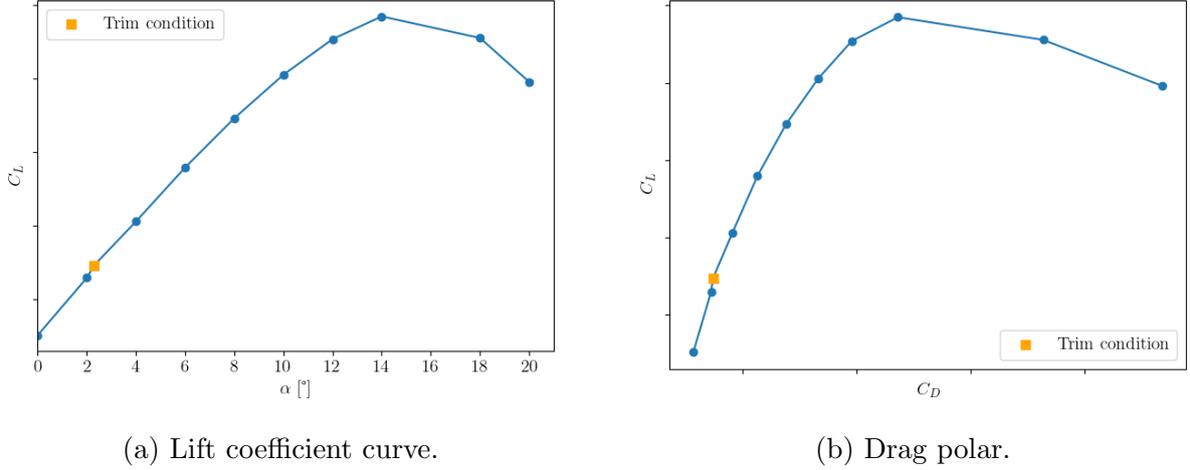
Table 4: Domain size study results

Chords	Nr. elements	Error C_L (%)	Error C_D (%)
5	467,200	-	-
10	496,400	0.23	8.35
15	511,000	1.14	1.89
20	525,600	0.02	0.93
25	540,200	0.10	0.73

It can be observed that the domain size has a bigger impact on the drag than on the lift coefficient. It is also possible to conclude that for 20 chords, the errors for both coefficients are under 1% which was considered enough for this work. Thus, this was the domain size used to perform the aerodynamic analysis.

3.5 Cruise performance

The pressure distribution over the surface can be observed in Figure 9a for an angle of attack of 2.3° , which was found to be the one that produced the prescribed wing lift coefficient for cruise conditions. It is possible to observe that on the upper surface, pressure is higher closer to the leading edge and in the winglet region and in this condition, the flow remains attached to the wing. Figure 9b, on the other hand, shows a flow distribution for a more extreme angle of attack, where it is clearly possible to see the flow separation on the suction side of the wing. In both figures it is possible to see the trailing edge vortex forming.


 Figure 10: Wing performance for angles of attack between 0° and 20° .

To observe the behavior of the wing with angle of attack, a study was performed and the results presented in Figure 10a. It is possible to see that C_L and C_D increase continuously with angle of attack until around $\alpha = 14^\circ$ and the variation is linear until around $\alpha = 8^\circ$, after which boundary layer separation starts to occur. In the linear region, $C_{L\alpha} = 0.075/^\circ$. A drag polar is also presented in Figure 10b, where efficiency can be analyzed. In this case, it would be higher for the smallest angle in the tested range, of 0° , and the maximum would possibly be for negative angles of attack, outside the tested range, decreasing with increasingly higher angles of attack. The orange square represents the trim operating condition, and it is possible to see that the wing is not operating at its maximum efficiency.

4 WING OPTIMIZATION

4.1 Optimization problem formulation

The formulation of the full optimization problem in standard form and the bounds considered for each design variable are presented in Table 5.

Table 5: Formulation of the full optimization problem in standard form.

		Quantity	Lower bound	Upper bound	Units
minimize	C_D	1	-	-	-
w.r.t.	α	1	0	15	$^\circ$
	γ (twist)	7	-15	15	$^\circ$
	c (chord)	8	15	150	%
	Λ (sweep)	1	0	10	$^\circ$
	airfoil shape	24	-0.05	0.05	m
subject to	$C_{L_{cruise}} = C_{L_{prescribed}}$	1	0.8932	0.8932	-
	$S_{proj} = S_{prescribed}$	1	2.1691	2.1691	m^2
	t (thickness constraint)	24	90	200	%

An FFD box with dimensions $12 \times 8 \times 2$ (streamwise, spanwise and vertical direction, respectively) and a reference axis at 25% of the chord will be used. The parametrizations

for the different variables will be done as discussed in Section 2.3: Twist will be associated with a rotation of each section around the reference axis in the spanwise direction, chord will be parametrized with a scale relative to its initial value in the streamwise and vertical direction (with the same value for both) in relation to its initial value, sweep will be parametrized with a displacement in the streamwise direction of every reference axis point in a linear way followed by a rotation of each control point around the vertical direction and airfoil shape will be parametrized with displacements of the local FFD points in the vertical direction. As discussed before, root twist will be kept at 0° so it will only have seven design variables.

Most limits were defined in such a way that the optimizer would have enough freedom. However, in some cases, like the airfoil shape and the chord lower bound tighter limits had to be imposed to avoid further problems in the volume mesh perturbation resulting in an unusable grid. The thickness constraints were set to span the whole wing and enforced at four spanwise points and six streamwise points. They are important mostly for the airfoil shape optimization and ideally, the wingbox for the required wing structure would be used to set them. As this work deals with the uncoupled aerodynamic shape optimization, it was arbitrated that, at each point, the thickness could not be less than 90% of the initial value.

Regarding the convergence criteria, a successful optimization is finished if either the convergence accuracy is smaller than $1e-6$ or the number of iterations reaches 500. It was verified that on successful optimizations the desired convergence accuracy criterion was met well before the maximum number of iterations.

4.2 Initial geometry characterization

The optimization procedure requires the definition of an initial geometry. That could be the Tekever AR5 current wing, however, given the parametrization method chosen (using FFD volumes), it would be hard to accurately define the design variables in such a way that they would match meaningful wing design parameters. For this reason, two simpler initial geometries were created: one, shown in Figure 11a which is a planification of the Tekever AR5 wing, which had the winglet removed and was then subjected to three main modifications: wingspan extension to match the projected wingspan of the original wing; a change in the trailing edge sweep of the outer section in order to keep the same projected area as the original wing; a removal of the existing twist; the other, shown in Figure 11b is a simple rectangular wing with the same projected wingspan and area as the Tekever AR5 wing and where the original airfoil was replaced with a symmetric NACA 4-series airfoil with the same thickness-to-chord ratio as the original wing. Figure 11 also shows the control points used with each geometry (blue circles) and the reference axis (shown in red).

The rectangular wing utilized a simple parallelepiped-shaped box, while the simplified Tekever AR5 wing employed a more complex FFD box that conforms to the surface geometry. This choice was made due to the non-symmetric nature of the airfoil. In both cases, the reference axis is positioned at 25% and 50% in the streamwise and vertical directions, respectively.

A mesh with a similar refinement level of the one obtained in 3.3 and around 900,000 elements was used for both wings. Given the geometric differences on the planform of both wings, they are not expected to create the same lift at the same angle of attack.

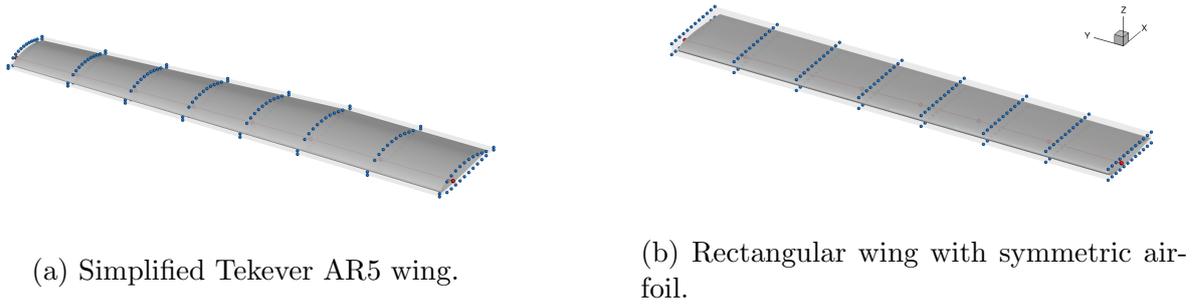


Figure 11: Wings used as a starting point for the optimization problem.

For this reason, an optimization considering only angle of attack as a design variable was performed. Given that the optimizer has only one degree of freedom and one constraint (a fixed lift coefficient), it will only be able to satisfy the constraint, and further drag minimization will not be possible. This allows, however, to find the trim angle of attack needed for both wings to maintain flight at the prescribed $C_{L_{wing}}$ and the corresponding C_D .

4.3 Results for the rectangular wing

The results obtained departing from the rectangular wing and considering different individual design variable optimization subproblems are presented in Table 6 and will be briefly analyzed next.

Table 6: Optimization results for the rectangular wing as starting geometry.

Case		α	C_D
Starting geometry		10.25°	Reference
Twist	Linear variation	12.89°	-2.60%
	3 FFD sections	11.57°	-3.06%
	7 FFD sections	11.86°	-3.21%
Chord	Linear variation	10.05°	-4.15%
	4 FFD sections	10.02°	-5.15%
	8 FFD sections	10.03°	-5.21%
Twist (7 sections) + Chord (8 sections) + Sweep		10.03°	-5.18%
Airfoil	Constant (24 Control points)	6.46°	-6.17%
	Variable (24 × 5 Control points)	5.82°	-8.55%

Twist optimization

The drag coefficient reduces with the increased degrees of freedom provided to the optimizer, however, differences between three and seven FFD sections are very small. Furthermore, less FFD sections usually led to smoother geometries, which may be desirable. The linear twist variation also poses itself as an interesting option, given that the drag reduction difference between it and the best option is 0.61% and it may be easier to manufacture.

Figure 12 compares the lift distribution of the obtained solutions. It can be seen that the distributions tend to the elliptical one as the number of degrees of freedom increase, but do not match it, having a slightly lower variation of the twist angle across the wingspan. A reason for this is the fact that separation started being observed near the trailing edge for such high incidence angles. This is an effect that could not have been predicted with optimization considering inviscid flow, to which the elliptical distribution is optimal.

It is also possible to note that the optimized solution for three and seven FFD sections is in fact very close, explaining the small differences in the drag coefficient observed. It should be noted that for the case with three FFD sections, the optimizer required only 18 function evaluations, whilst for the seven FFD sections case it required 79, representing a very significant difference in computational cost.

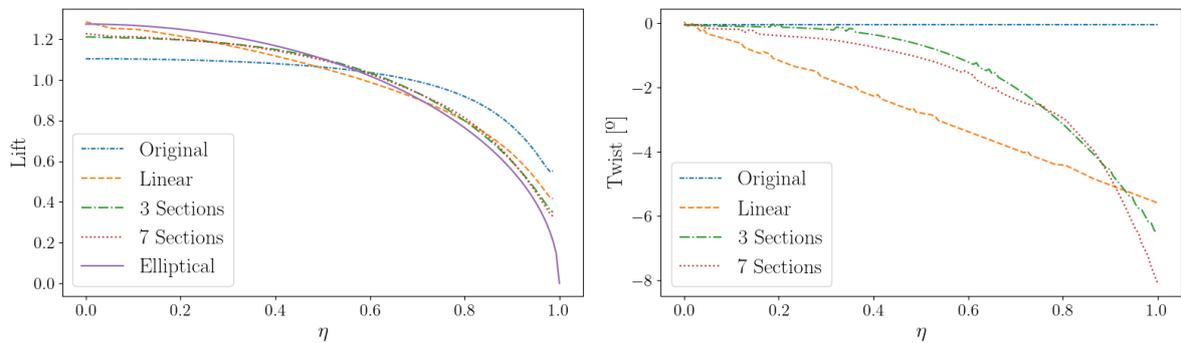


Figure 12: Comparison between lift distribution for the different twist distributions.

Compared to other variables, it is also possible to observe that the twist distribution let the angle of attack be higher than the initial geometry. This is expected, given that all the obtained twist distributions applied a washout, reducing the incidence monotonically across the whole wingspan and, thus, reducing the angle of attack of those sections.

Chord optimization

A similar pattern to the twist distribution was observed regarding the number of degrees of freedom, with the results improving with their increment but with the four and eight FFD section cases yielding very similar results. Here, the difference between the linear variation and the one with eight FFD sections is 1.06%.

The optimized wings feature lower drag coefficients than the ones obtained with twist distribution. One factor that may contribute to this is the additional degree of freedom that chord has at the root, despite having a constraint on the projected area to allow a fair comparison between different geometries. The linear chord variation is effectively equivalent to a trapezoidal wing with taper $\lambda = 0.358$.

Lift and chord distributions for the three cases are shown in Figure 13, where it can be seen that the distribution is indeed closer to elliptical than the one obtained with twist optimization. Regarding the four and eight FFD sections cases, similarly to the twist case, the lift distributions are very close to each other.

Figure 14 compares the suction side pressure distribution of the optimized wing with eight FFD sections with the original one. The elliptical shape and lift distributions ob-

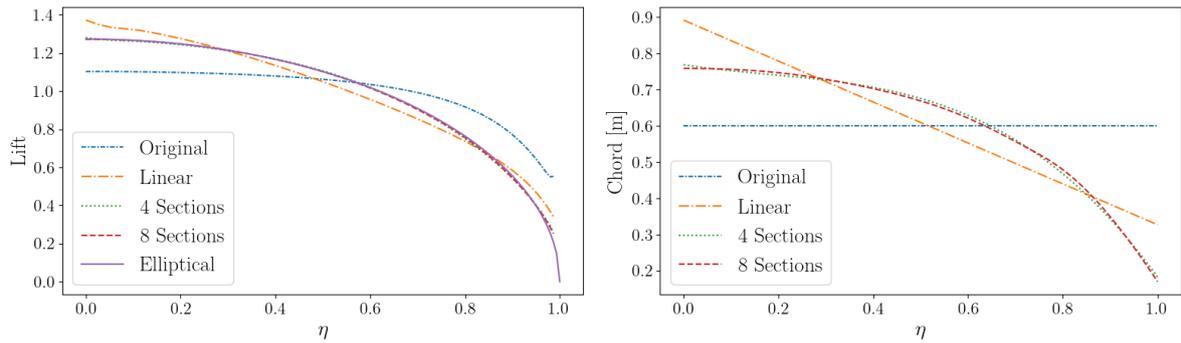


Figure 13: Comparison between lift distribution for the different chord distributions.

tained are evident, with the most visible differences near the wingtip.

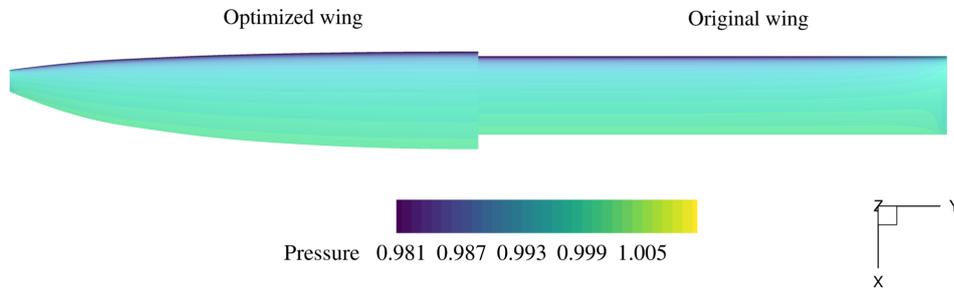


Figure 14: Comparison of the pressure distribution in the suction side between the original and the optimized wing.

Twist + chord + sweep optimization

A combined case with all global design variables was also considered and a very similar drag coefficient to the chord optimization case with eight FFD sections was obtained.

Airfoil optimization

The airfoil optimization considering a constant airfoil on the three-dimensional wing, allowing the capture of induced drag effects, yielded the second best result for drag reduction. Starting from a symmetric NACA 4-series airfoil, the optimizer took 113 function calls and 28 gradient calls to converge to a cambered airfoil, where thickness was mostly taken to the minimum value allowed by the constraint. The obtained airfoil profile is outlined in orange in Figure 15a. With this, the required angle of attack for the prescribed wing lift coefficient was reduced by 3.8° . Another significant change was the distribution of pressure and viscous drag, with the viscous drag only representing 10.5% of the total drag of the optimized geometry, compared to 21.5% on the original wing. Figures 15b and 15c compare the obtained pressure distribution for both airfoils at a section at the middle of the wing and near the tip, respectively. Considering the middle section, it can be seen that the pressure peak on the suction side was greatly decrease on the optimized airfoil, with a smoother recuperation along the chord.

The highest drag reduction was obtained for a variable airfoil across the wing. For this case, a different FFD box was used with only five sections in the wingspan direction in

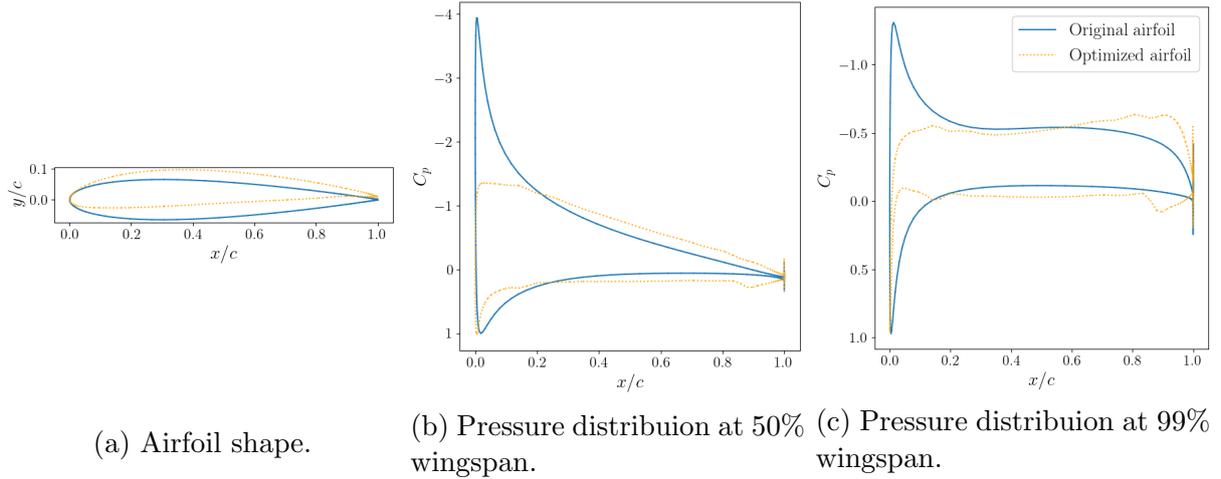


Figure 15: Comparison between the original airfoil and the optimized one.

order to reduce the number of design variables and thus, the computational time needed. Figure 16 shows the airfoils obtained for a section near the root, a section at the middle of the wing and a section near the tip. As expected, the sections near the root and at the middle of the wing are similar, given that a symmetry boundary condition was applied at the root and wingtip effects are not felt yet at those regions, although it has a slightly higher camber near the root. The most notable difference occurs near the wing tip, where the airfoil closely matches the original one, with barely any camber.

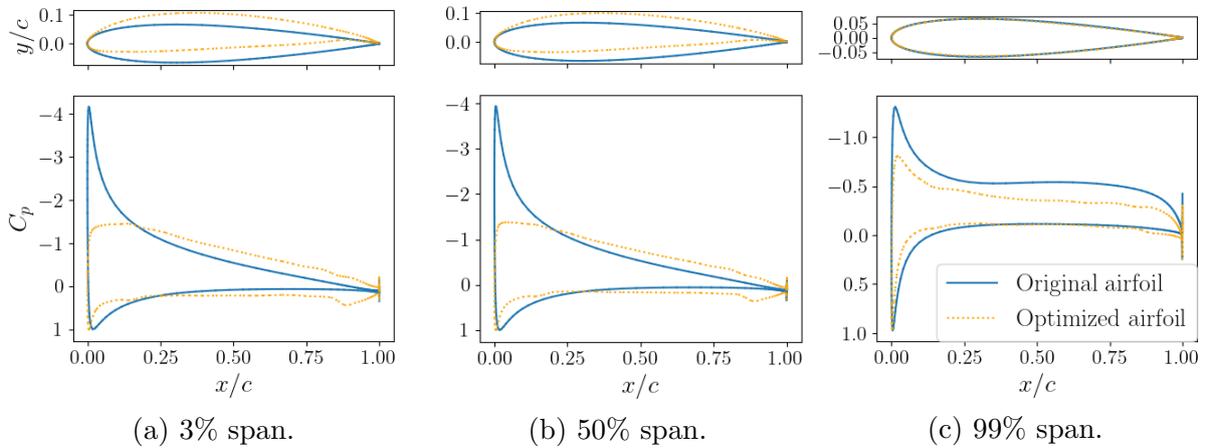


Figure 16: Comparisof the pressure distribution in the suction side between the original and the optimized wing at three sections.

To reach the optimized wing, 128 function calls and 35 gradient calls were needed, representing an increase of only 13% and 25% to the constant airfoil case. It should be noted, however, that the gradients naturally took longer to compute in the variable airfoil case, given the higher number of design variables.

4.4 Results for the simplified Tekever AR5 wing

The summarized results of the optimization using the simplified Tekever AR5 wing as starting geometry geometry are found in Table 7.

Table 7: Optimization results for the simplified Tekever AR5 wing as starting geometry.

Case	α	C_D
Starting geometry	4.03°	Reference
Twist	Linear 2 sections	3.96° -0.39%
	7 FFD sections	4.56° -0.80%
Chord (Linear 2 sections)	4.07°	-0.11%
Twist (7 sections) + Chord (2 linear sections)	4.09°	-1.16%
Airfoil (24 × 8 Control points)	4.59°	-2.54%

Twist optimization

Two cases were defined: A linear variation and a variation with full freedom on the seven FFD sections. Given the the nature of the original geometry, that can be defined by two separate sections, the linear case was defined in two separate sections. To define them, the optimizer was able to change the twist angle at the junction between both sections and at the wingtip. From these values, two separate linear distributions are defined.

It is clear that the initial geometry is much closer to the elliptical lift distribution than the rectangular wing and for this reason, the drag reductions obtained are much smaller. In both scenarios, whether we consider a linear twist distribution defined in two linear sections or seven FFD sections, the optimizer successfully approached an elliptical lift distribution. Notably, the case involving seven FFD sections resulted in nearly twice the drag reduction compared to the linear twist distribution defined in two sections. This significant improvement can be observed in Figure 17, where the lift distribution of the seven FFD sections case is closer to elliptical.

Chord optimization

The parametrization for the chord followed a similar approach to that of twist. However, it was adjusted to ensure a constant taper at the inner section by enforcing the same scaling factor applied to the first two FFD sections, representing the first degree of freedom of the optimizer. The second determined the scaling factor at the wingtip and a linear distribution was then defined between the junction between sections and the wingtip. Due to the constraint imposed by the projected area, the optimizer only had one effective degree of freedom, which determined the taper, λ , for the outer section of the wing. This case was the only configuration tested, in order to maintain the original wing design philosophy. The optimizer produced a different geometry, with the final scale factors being 1.09 and 0.76, but the resulting drag reduction for this particular case was minimal.

Twist + chord optimization

A combined optimization case with full freedom to the seven twist design variables and the taper of the outer section was also considered and, unlike the rectangular wing, the results were better than any of the singular cases, obtaining a drag reduction of more than 1%. The obtained scale factors for chord distribution were 1.13 and 0.67, and a twist distribution was also applied, with its lowest value, at the wingtip, being -4.14° . This

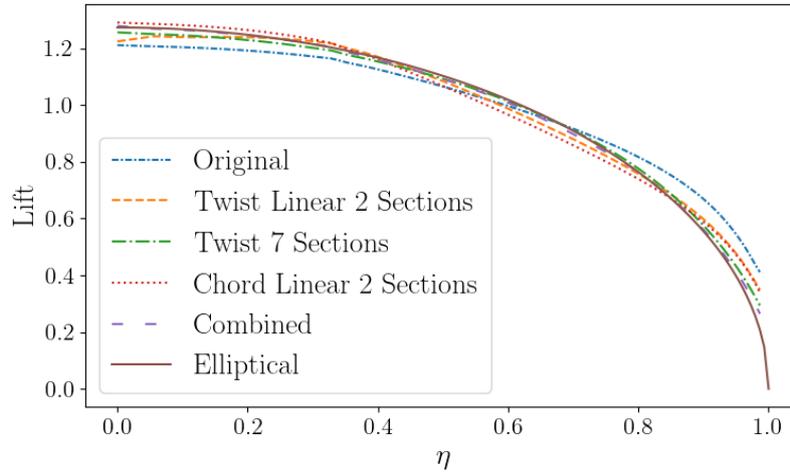


Figure 17: Comparison between lift distribution for the different parametrizations.

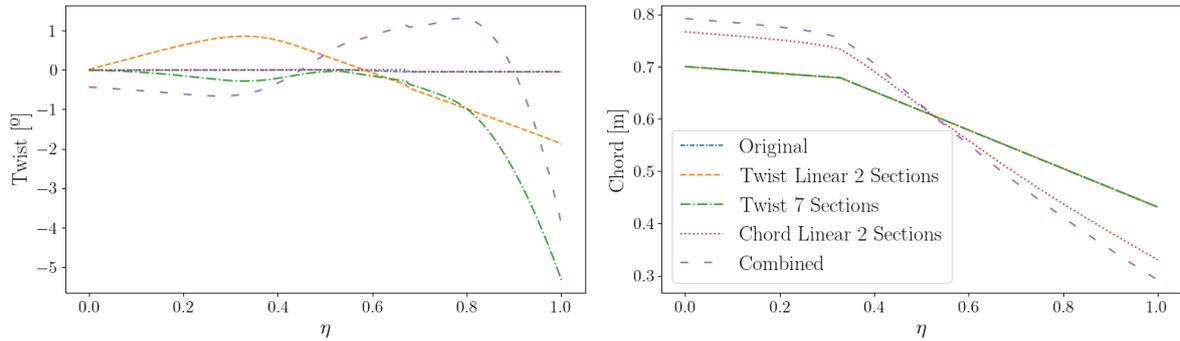


Figure 18: Comparison between chord and twist distribution for the different parametrizations.

results in a lift distribution almost elliptical, as Figure 17 shows. In Figure 18, it is possible to see the twist and chord distribution obtained for this and the other parametrizations tested, starting from the simplified Tekever AR5 wing geometry.

Airfoil optimization

Finally, a variation in airfoil shape along the wingspan was also optimized. Similar to the rectangular wing case, this optimization yielded the greatest reduction. However, the reduction achieved was significantly smaller than that of the rectangular wing, which was expected considering that the starting profile for this case was already optimized for the specified flight condition.

Figure 19 provides a comparison between the original airfoil profile and the optimized one at three different sections along the wingspan. The comparison reveals that the modifications in the initial sections mainly involved a decrease in thickness and a slight reduction in camber. Interestingly, at the wing tip, the optimizer converged to a symmetric airfoil, resembling the case of the rectangular wing with a symmetric airfoil geometry as a starting geometry.

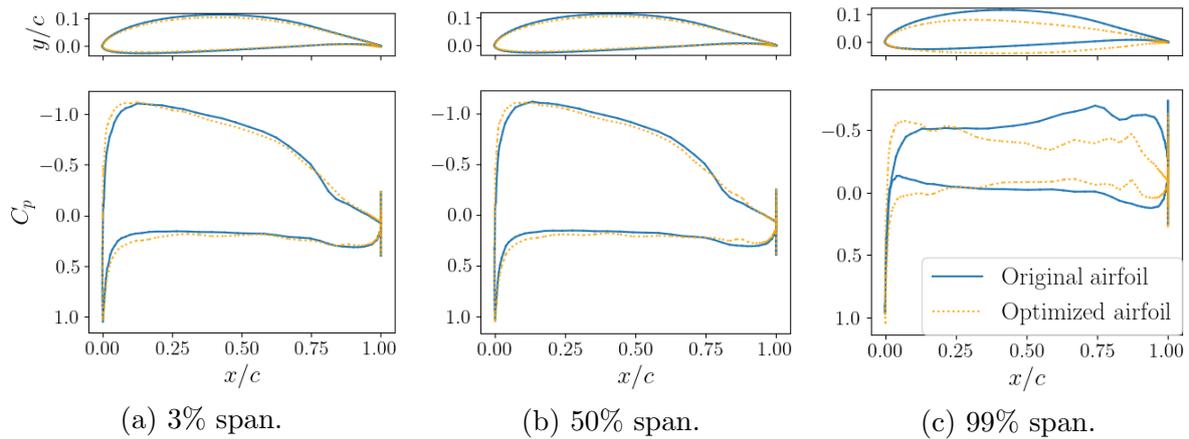


Figure 19: Comparison of the pressure distribution in the suction side between the original and the optimized wing at three sections.

5 CONCLUSIONS

This work presents a study on the optimization of two different starting geometries, namely the rectangular wing with a symmetrical airfoil and the simplified Tekever AR5 wing, using different parametrizations. The results demonstrate notable drag reductions of up to 8.55% for the rectangular wing and 2.54% for the simplified Tekever AR5 wing.

Additionally, most of the optimized lift distributions closely approximated the elliptical one. The use of two starting geometries in this study highlights the potential for optimizing both naive geometries and geometries obtained from extensive design processes, thereby showing capabilities for both refinement of existing designs as well as the creation of new designs from scratch considering a specified flight condition.

ACKNOWLEDGEMENTS

The authors would like to thank Tekever for supporting the production of this article by offering all the needed data to successfully produce the analyses here shown. This work was supported by FCT, through IDMEC, under LAETA, project UIDB/50022/2020.

REFERENCES

- [1] K. P. Valavanis and G. J. Vachtsevanos. *Handbook of Unmanned Aerial Vehicles*, volume 1. Springer. ISBN 978-90-481-9706-4.
- [2] V. Prisacariu. The history and the evolution of uavs from the beginning till the 70s. *Journal of Defense Resources Management (JoDRM)*, 8(1):181–189, 2017.
- [3] J. Vos, A. Rizzi, D. Darracq, and E. Hirschel. Navier–Stokes solvers in European aircraft design. *Progress in Aerospace Sciences*, 38(8):601–697, 2002. doi:10.1016/S0376-0421(02)00050-7.
- [4] P. E. Rubbert. CFD and the changing world of airplane design. In *19th Congress of the International Council of the Aeronautical Sciences*, volume 19, pages LVII–LVII,

- Anaheim, USA, September 1994. American Institute of Aeronautics and Astronautics.
- [5] S. Defoort, M. Méheut, B. Paluch, R. Liaboeuf, R. Murray, D. C. Mincu, and J.-M. David. Conceptual design of disruptive aircraft configurations based on high-fidelity oad process. In *Aviation Technology, Integration, and Operations Conference*, page 3663, Atlanta, USA, june 2018.
- [6] R. Haimes and M. Drela. On the construction of aircraft conceptual geometry for high-fidelity analysis and design. In *50th AIAA Aerospace sciences meeting including the new horizons forum and aerospace exposition*, page 683, Nashville, USA, january 2012.
- [7] P. D. Bravo-Mosquera, H. D. Cerón-Muñoz, G. Díaz-Vázquez, and F. M. Catalano. Conceptual design and CFD analysis of a new prototype of agricultural aircraft. *Aerospace Science and Technology*, 80:156–176, 2018.
- [8] Drone Industry Insights. Global drone market report 2022-2030, 2022.
- [9] J. R. Martins. Aerodynamic design optimization: Challenges and perspectives. *Computers & Fluids*, 239:105391, 2022. doi:10.1016/j.compfluid.2022.105391.
- [10] C. A. Mader, G. K. Kenway, A. Yildirim, and J. R. Martins. ADflow: An open-source computational fluid dynamics solver for aerodynamic and multidisciplinary optimization. *Journal of Aerospace Information Systems*, 17(9):508–527, 2020.
- [11] G. K. Kenway, C. A. Mader, P. He, and J. R. Martins. Effective adjoint approaches for computational fluid dynamics. *Progress in Aerospace Sciences*, 110:100542, 2019. doi:10.1016/j.paerosci.2019.05.002.
- [12] G. Kenway, G. Kennedy, and J. R. Martins. A CAD-free approach to high-fidelity aerostructural optimization. In *13th AIAA/ISSMO multidisciplinary analysis optimization conference*, page 9231, Fort Worth, USA, September 2010.
- [13] N. Wu, G. Kenway, C. A. Mader, J. Jasa, and J. R. Martins. pyOptSparse: A python framework for large-scale constrained nonlinear optimization of sparse systems. *Journal of Open Source Software*, 5(54):2564, 2020. doi:10.21105/joss.02564.
- [14] R. A. McDonald and J. R. Gloudemans. Open Vehicle Sketch Pad: An open source parametric geometry and analysis tool for conceptual aircraft design. In *AIAA SciTech 2022 Forum and Exposition*, page 4, San Diego, USA, january 2022.
- [15] D. Poirier, S. Allmaras, D. McCarthy, M. Smith, and F. Enomoto. The CGNS system. In *29th AIAA, Fluid Dynamics Conference*, page 3007, Honolulu, USA, June .
- [16] C. Rumsey, B. Wedan, T. Hauser, and M. Poinot. Recent updates to the CFD general notation system (CGNS). In *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, page 1264.

- [17] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331. doi:10.1002/nme.2579.
- [18] N. Secco, G. K. W. Kenway, P. He, C. A. Mader, and J. R. R. A. Martins. Efficient mesh generation and deformation for aerodynamic shape optimization. *AIAA Journal*, 59(4), 2021. doi:10.2514/1.J059491.
- [19] M. Z. M. Shah, B. Basuno, and A. Abdullah. Comparative study on several type of turbulence model available in Ansys-Fluent software for Onera M6 wing aerodynamic analysis. *Journal of Advanced Mechanical Engineering Applications*, 1(1):9–19, 2020.
- [20] P. Eliasson. Influence of turbulence modelling and grid resolution in computations of the dpw7 crm configuration. In *AIAA Aviation Forum and Exposition*, page 4091, San Diego, USA, june 2023.
- [21] L. Kral. Recent experience with different turbulence models applied to the calculation of flow over aircraft components. *Progress in Aerospace Sciences*, 34(7-8):481–541, 1998. doi:10.1016/S0376-0421(98)00009-8.
- [22] J. G. Coder, T. H. Pulliam, D. Hue, G. K. Kenway, and A. J. Sclafani. Contributions to the 6th AIAA CFD drag prediction workshop using structured grid methods. In *55th AIAA Aerospace Sciences Meeting*, page 960, Grapevine, USA, january 2017.
- [23] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th aerospace sciences meeting and exhibit*, page 439, Reno, USA, january 1992.
- [24] L. Eca, F. Pereira, and G. Vaz. Viscous flow simulations at high reynolds numbers without wall functions: Is y^+ approximately 1 enough for the near-wall cells? *Computers & Fluids*, 170:157–175, 2018. doi:10.1016/j.compfluid.2018.04.035.
- [25] MDO Lab. *MACH-Aero Documentation*. MDO Lab, University of Michigan, USA, 2022.
- [26] MDO Lab, University of Michigan, USA. *pyGeo Documentation*. MDO Lab, 2022.
- [27] D. W. Zingg, M. Nemec, and T. H. Pulliam. A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, 17(1-2): 103–126, 2008.
- [28] Z. Lyu, Z. Xu, and J. Martins. Benchmarking optimization algorithms for wing aerodynamic design optimization. In *Proceedings of the 8th International Conference on Computational Fluid Dynamics, Chengdu, Sichuan, China*, volume 11, page 585. Citeseer, 2014.
- [29] N. Bons. *High-fidelity Wing Design Exploration with Gradient-based Optimization*. PhD thesis, University of Michigan, USA, 2020.

- [30] H. Koyuncuoglu and P. He. CFD based multi-component aerodynamic optimization for wing propeller coupling. In *AIAA SciTech 2023 Forum and Exposition*, page 1844, National Harbor, USA, january 2023.
- [31] G. Yang and A. Da Ronch. Aerodynamic shape optimisation of benchmark problems using SU2. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0412, 2018.
- [32] Y. Yu, Z. Lyu, Z. Xu, and J. R. Martins. On the influence of optimization algorithm and initial design on wing aerodynamic shape optimization. *Aerospace Science and Technology*, 75:183–199, 2018. doi:10.1016/j.ast.2018.01.016.
- [33] D. Kraft. A software package for sequential quadratic programming. *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt*, 1988.
- [34] Tekever. Tekever>Meet Our UAS Models>AR5. URL <https://http://www.tekever.com/models/ar5/>.
- [35] MDO Lab. *ADFlow Documentation*. MDO Lab, University of Michigan, USA, 2022.