Airframe Design

Madsen Evans

December 18, 2024

1 Nomenclature

- numerical optimization program: a program which maximizes or minimizes an objective function within a set of constraints
- objective function: a function which takes a set of design variables and returns a value
- design variables, δ : the variables that the numerical optimization program can choose to find the optimum
- filter matrix, W : a matrix which is used to make each chord length depend in part on the other chord lengths nearest it.
- monotonicity: the quality of a wing where the chord length decreases from root to tip
- mean aerodynamic chord, c_{mac} : $c_{mac} = \frac{2}{S}\int_0^{b/2} c^2\,dy$, where $S = 2\int_0^{b/2} c\,dy$ It is a chord-weighted average chord.
- wing span, b : the lateral extent of the wing (both sides)
- wake vortex: a linear vortex that trails behind a wing (vortex filaments)
- downwash: downward moving air behind a wing
- upwash: upward moving air behind a wing

2 Introduction

As described in Engineering Design Optimization by Joaquim R. R. A. Martins and Andrew Ning, with the increased capacity of computers, engineers have the opportunity to make use of numerical optimization programs to find optimal designs. Using numerical optimization to find an optimal design, however requires additional knowledge to incorporate optimization into the engineering process. Numerical optimization programs work in a variety of ways, but they all require three basic inputs. The first in an objective function. An objective function is a function which returns a value which the numerical optimization program seeks to maximize or minimize. The second input consists of constraints. Constraints define the boundaries of the search space. And thirdly, a starting point must be provided to the optimizer.

An objective function must be chosen wisely such that the objective reflects the design goals for a design. For example, it would likely not be advisable to optimize a wing to maximize loading cycles as it would probably create an unwieldy wing. Similarly, the constraints must be chosen in such a way that they reflect the requirements of the design without excessively shrinking the search space. Additionally, selecting the starting point for the optimizer will have an effect on where it will converge as well as how quickly it will converge.

The aim of this project is to practice optimization by optimizing chord lengths along a wingspan. The wingspan is 8.0 m, oriented at a pitch angle of 5°, flying at a speed of 1 m/s, and has no sweep, twist, or dihedral. Additionally, the quarter chords of all sections are aligned. It is constrained to carry at least 1.7 N, and the objective is to minimize the dimensional drag.

3 Methods

For this project, the Julia packages SNOW.jl and VortexLattice.jl were used for the optimizer and the simulator, respectively.

A function that takes a vector of chord lengths was used to create a wing that satisfies the physical design requirements mentioned in the Introduction. That wing could then be run through VortexLattice to retrieve its coefficient of drag, which could then be reconstructed into its dimensional drag for the objective function.

The starting point provided to the optimizer was chosen in a hope to influence the end result as little as possible. The starting point was a wing with each chord segment set to a length of 0.1 meters.

During initial versions of the optimization code, the optimizer would return jagged wings with chord lengths of zero. This is because having jagged wings lets the vortex lattice method calculate a localized upwash which reduces the drag on the wing (Conlan-Smith et al.). However, this does not accurately reflect reality, so a solution had to be devised.

Borrowing from Conlan-Smith et al.: a filter matrix (W) was applied to the design variables (δ) generated by the optimizer before they were fed to the simulator. In effect, this would make every chord a function of those nearest to it and removed the jaggedness of the wing.

$\tilde{\delta} = W\delta$ where $\tilde{\delta}$ is the vector of chord lengths used to build the wing, δ is the vector of design variables generated by the optimizer. W is such that:

$W_{ij} = \frac{1}{\sum_{k=1}^{N_s} w_{ik}} w_{ij}$ where $w_{ij} = max[0, R - d_{ij}]$ where R is a filter radius.

Sadly, upon applying this method, VortexLattice.jl was unable to correctly assemble the wings. The panels it generated from the provided chord lengths folded in on themselves creating impossible wing geometries which the optimizer exploited. An example of such a wing is shown in Figure 1.

Following that attempt, a monotonicity constraint was enforced.

4 Results

The results of the wing using the filter matrix are shown in Figure 1. The geometries of the wing created by VortexLattice.jl fold back on themselves, creating an impossible wing shape.

Figure 1: The result from the optimizer using the a filter matrix with 10 design variables.

Enforcing a monotonicity constraint yields an operational optimizer that works well up to 35 design variables. Trying to get more chord segments leads to the optimizer exceeding its maximum number of iterations. The right half of a wing designed by the optimizer using 34 chord segments is shown in Figure 2. It is clear that the taper from root to tip is not linear. A visual inspection reveals some similarity in shape to the famous elliptical wing of the Supermarine Spitfire. To assess this similarity with a more analytical method: Figure 3 compares the lift distribution of the optimized wing to an elliptical lift distribution.

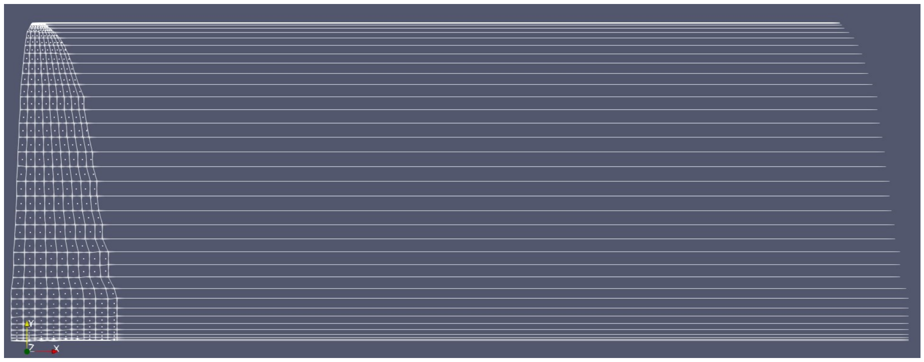

Figure 2: The result from the optimizer using the monotonicity constraint and 35 equally spaced span-wise chord segments.

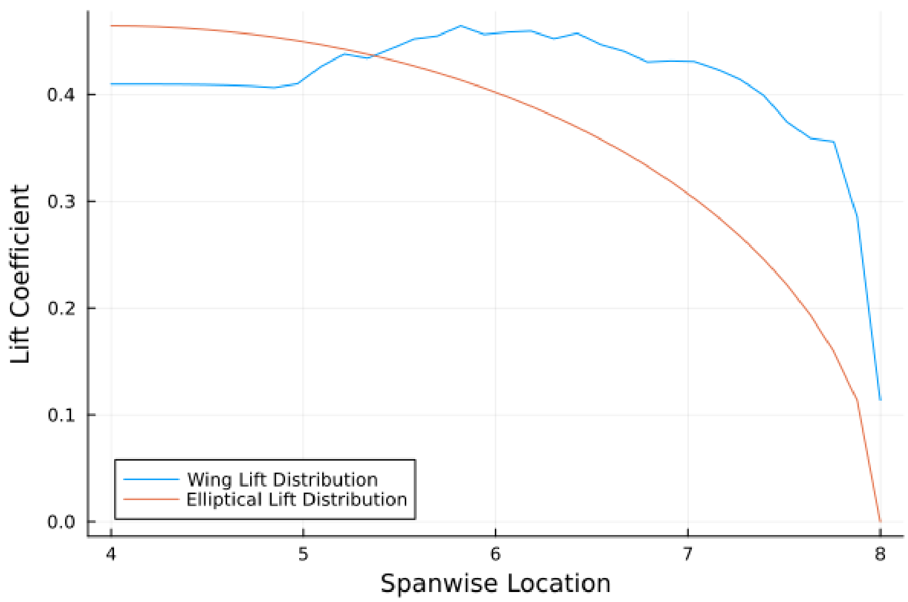

Figure 3: The coefficient of lift for each span-wise segment compared to an elliptical lift distribution with the same maximum lift coefficient.

In addition to creating a successful optimization program, one of the objectives of this assignment is to experiment with its limitations. To this end, the lift constraint was increased from 1.7 N to 17 N. Doing this, the optimizer was unable to converge, even for a set of 5 chord segments.

Finally, the tolerance of the optimizer was changed. The default tolerance is 1e-6. When using the default tolerance, the optimizer runs in 52 iterations and reports that the objective value—the drag—was 2.827e-2 N. Increasing the tightness of the tolerance to 1e-7 made it so that the optimizer was unable to converge. However, what it did return, was a wing with very nearly the same chord lengths and lift distribution as the optimizer with the default tolerance. And the objective came out to be within 0.0022% of the default's. Decreasing the tightness of the tolerance to 1e-5 resulted in a wing that once again had no visual differences to the wing generated with the default tolerance. The looser tolerance did however mean that the optimizer only had to run 35 iterations. And the objective was within 0.0002% of the default tolerance.

Loosening the tolerance to 1e-1 meant that the optimizer ran in 19 iterations and reported a drag within 0.081% of the drag reported by the default tolerance.

5 Discussion

From Flight Vehicle Design by Andrew Ning, we know that the lift distribution which produces the least induced drag is an elliptic lift distribution. Figure 3 shows that the optimizer finds an optimum that roughly follows an elliptical lift distribution. However, there are some differences. The main difference between the two distributions is that that the maximum lift coefficient occurs at different locations on the wing. On the generated wing, it is almost halfway between the root and the tip as opposed to at the root, as on the elliptical lift distribution.

This discrepancy between the generated optimum and the calculated optimum could be a result of the constraints imposed on the optimizer. Constraints reduce the search space available to the optimizer, and whenever the optimum solution provided bumps up against the constraints, it is evident that a more optimal answer could be available should the constraints be removed. And it is clear that the monotonicity constraint is active on this wing, meddling with the ability of the optimizer to find a better solution.

The lack of variance in the results even with the different levels of tolerance in the optimizer shows that the result is quite steady. This bodes well for those using a vortex lattice method as a first level of optimization in a larger engineering project because running it quickly, at a lower fidelity, will still return a result quite similar to the results it would return with a tighter tolerance.

6 Resources

Conlan-Smith, C., Ramos-García, N., Sigmund, O., & Andreasen, C. S. (2020). Aerodynamic Shape Optimization of Aircraft Wings Using Panel Methods — AIAA Journal. AIAA Journal, 58(9). Aerospace Research Central. https://doi.org/10.2514/1.J058979

Joaquim R. R. A. Martins and Andrew Ning. Engineering Design Optimization. Cambridge University Press, 2021. ISBN: 9781108833417.

Ning, A. (2022). Computational Aerodynamics. (Original work published 2020)

Ning, A. (2024). Flight Vehicle Design [Review of Flight Vehicle Design]. (Original work published 2018)