Master of Aerospace Engineering

Research Project

Blended Wing Body Wingbox Design with Aeroelasticity Constraints

S2 Final Report

Author: João Cruz Matos

Due date: 01/07/2019
Submission date: 24/06/2019

Starting date of the project: 29 Jan 2019 **Duration**: 14 Months

Tutors: J. Morlier, S. Coniglio, J. Mas Colomer

Declaration of Authenticity

This assignment is entirely my own work. Quotations from literature are properly indicated with appropriated references in the text. All literature used in this piece of work is indicated in the bibliography placed at the end. I confirm that no sources have been used other than those stated.

I understand that plagiarism (copy without mentioning the reference) is a serious examinations offence that may result in disciplinary action being taken.

Date	Signature
24/06/2019	João Matos

Abstract

This report is a record of the progress of a research project on a blended wing body wingbox design with aeroelasticity constraints. The project consists in designing a wingbox through topology optimization and with multidisciplinary optimization add to the designing process the aerodynamic forces applied on the wing. This wingbox design will be applied to the an innovative aircraft design, a blended wing body design, that is predicted to improve the fuel consumption in about 10% compared to the conventional wide body aircraft.

For the multidisciplinary optimization the platform OpenMDAO will be used, so the development of a topology optimization implementation in Python was the main goal for this semester. The implementation in Python uses the SIMP approach to topology optimization, as a solver it uses the Method of Moving Asymptotes (MMA) and it has a variety of different approach methods to the boundary penalization implemented.

Keywords

Aerospace Engineering, Wingbox Design, Topology Optimization, SIMP, Multidisciplinary Optimization, Blended Wing Body Design

Contents

List of Figures

1 Introduction

This research project is on the topic of blended wing body wingbox design with aeroelasticity constraints. The project has a duration of one year and will be conducted as part of the second and third semester of the Master in Aerospace Engineering. The main goal of the project is to through topology optimization create a wingbox design that takes into account the aeroelasticity and apply it to an innovative design, a blended wing body design (BWB).

1.1 Objectives

The project is separated in five different steps all completely dependent on each other, so the work in each step can only start once the previous step has already been completed.

1. Develop a topology optimization implementation in Python.
2. Add wingbox boundary conditions to the implementation.
3. Build a MDO formulation to design a wingbox with the topology optimization code.
4. Add aeroelasticity constraints.
5. Apply it to a blended wing body design (BWB).

To achieve the project goal of designing an optimized wingbox structure for a BWB, an OpenMDAO formulation with multiple disciplines is needed. For the aerodynamics discipline it already exists an aeroelasticity toolbox created by J. Mas Colomer but for the structures discipline there is the needed to create a topology optimization code that optimizes a wingbox.

2 Semester 2 Work

The goals of the project for the semester were well defined:

1. State of the art.
2. Learn the basics about topology optimization.
3. Understand the topology optimization codes.
4. Develop a topology optimization code.
5. Add different methods to this code.

First a solid state of the art was needed, to find out what has been done in topology optimization and how to start this project. With the state of the art done there were still entry barriers to get into the topology optimization topic, so a lot of work was done to learn more about the topic. After knowing more about the topology optimization it was easier to understand the different implementation codes already developed on topology optimization.

The main goal for the semester was to develop a new topology optimization implementation in Python, this was possible from joining and translating some of the already developed codes. To the new developed implementation in Python more boundary conditions and different methods were added.

2.1 Context and Key Issues

To achieve the main goal for the semester some problems had to be overcome. The biggest issue was that to do multidisciplinary optimization with the platform OpenMDAO a topology optimization implementation in Python was needed because OpenMDAO only works in Python. Most of the already developed topology optimization implementations use MATLAB instead, including the work done in ISAE-Supaero on the topic by S. Coniglio.

The programming codes used in this project when developing the implementation in Pyhton were:

- A 99 line topology optimization code written in MATLAB - Sigmund, O.
- Efficient topology optimization in MATLAB using 88 lines of code - Andreassen, E., Clausen, A., Schevenels, M. et al.
- Generalized Geometry Projection - Coniglio, S.
- Topology optimization codes written in Python.

A easy code to understand and recommended for when first learning topology optimization is the 99 lines MATLAB code by Sigmund (2001). A good follow up for this code is the 88 lines MATLAB code by Andreassen et al. (2011), it is more efficient but also harder to understand. The third implementation on the list is by Simone Coniglio, where he compares three different methods and suggests a new one. This was the main code used when developing the Python implementation for the project.

Even though these three codes were all in MATLAB there is available a Python version of the 88 lines MATLAB code.

2.2 Theoretical Background

Before starting the project a state of the art analysis was performed to better understand what has already been done in the topic of topology optimization, not only to understand more about the topic but also to avoid redoing work that is available.

2.2.1 Topology Optimization

This semester's work focused mostly in topology optimization. Topology optimization is normally used in preliminary phases of the designs to predict the optimal material distribution within a given initial design space. According to Hayoung Chung (2019) topology optimization is a numerical method that computes an optimal structural layout for a set of objectives and constraints with the goal of getting a lighter structure that uses less material and will only have material in the most critical areas.

There are different formulations in topology optimization, in this project it will be used the Solid Isotropic Materials with Penalization (SIMP) method.

A classical example of a SIMP topology formulation is presented in Equation 2.1. The objective of this problem is to minimize the compliance.

$$\begin{cases} min_{\{x\}}: \ c = U^T F \\ \frac{V(x)}{V_0} \leq f \\ KU = F \\ 0 \leq x_{min} \leq x \leq 1 \end{cases} \tag{2.1}$$

Where c is the compliance, U is the global displacement vector, F is the force vector, $V(x)$ is the material volume, V_0 is the design domain volume, f is the constrained volume fraction, K is the global stiffness matrix, x is the vector of design variables (the density of each element) and x_{min} is a minimum density to avoid singularity.

This optimization problem can be solved using different methods like the Optimally Criteria and the Method of Moving Asymptotes (Svanberg, 1987). In the implementation it will only be used the Method of Moving Asymptotes (MMA).

2.2.2 SIMP

The Solid Isotropic Materials with Penalization (SIMP) method is a pixel based method. On this method the design variable x, the density of each finite element, is a continuous variable between 0 that represents void and 1 that represents fully filled with material.

According to Hayoung Chung (2019) because of the continuity and the bound, the optimization problem is well suited to employ gradient based optimization. However, the continuous density leads to an intermediate density that makes the identification of the result challenging. So these intermediate densities are penalized to get a better defined result.

The papers by Sigmund (2001) and by Andreassen et al. (2011) present and explain implementations with the SIMP method.

2.2.3 OpenMDAO

OpenMDAO is a platform developed by NASA (Gray et al., 2010). According to Hayoung Chung (2019) it is used to solve MDO problems in different fields of engineering because it can handle a large number of variables and disciplines. In this project it will be used to build a MDO formulation to design the wingbox through topology optimization.

This MDO formulation will also take into account the aerodynamic forces with some aeroelasticity constraints from the aeroelasticity toolbox developed by J. Mas Colomer.

2.3 Justification of the Potential Degree of Novelty

The topic of this research project is very relevant to the aeronautic industry due to two main aspects incorporated in the project:

- Aerostructural coupling - the design of the wingbox will take into account the interactions between the structure of the wings and the aerodynamic forces. These forces deform the original shape of the wing and this deformation will consequently change the value of the same aerodynamic forces. An optimization loop with both disciplines will result in a more efficient and accurate design of the wingbox.
- Blended wing body design - the wingbox design will be applied to an innovative design, a BWB. This will contribute to the innovation of civil aviation that in a long time has not seen any substantial changes in the designs of the aircraft.

Both of these are important steps to innovate the aviation industry making it a more efficient designing process and developing more efficient aircraft.

3 Numerical Implementation

The topology optimization implementation developed in Python uses the SIMP approach (Section 2.2.2). Two SIMP approach methods were implemented on the code: the Moving Morphable Components (MMC) (Guo et al., 2014) and the Generalized Geometry Projection (GGP) by Simone Coniglio. These methods differ mostly on their approaches to the penalizations of the intermediate densities around the defined components.

The formulation of the optimization implemented is presented in Equation 3.1

$$\begin{cases} min_{\{x\}} c = U^T K U = \sum_{e=1}^{N} (x_e)^p u_e^T k_0 u_e \\ \frac{V(x)}{V_0} \leq f \\ 0 \leq x_{min} \leq x \leq 1 \end{cases} \tag{3.1}$$

Where x is the vector of design variables, N is the number of elements, u_e is the element displacement vector and k_e is the element stiffness matrix

The optimizer used on the project to solve this optimization problem was the Method of Moving Asymptotes (Svanberg, 1987).

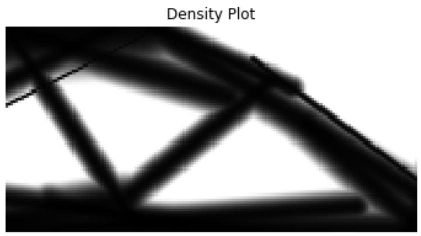
4 Results and Analysis

On the implementation the program plots the graphs with a rate of $plot\_rate = 100$ iterations. It plots the density plot (Figure 4.1a), the components plot (Figure 4.1b), the compliance (Figure 4.2) and the volume fraction (Figure 4.3). These graphs were plotted for a mesh of 160 x 80, a MBB beam, the GGP method and a volume fraction of 50%.

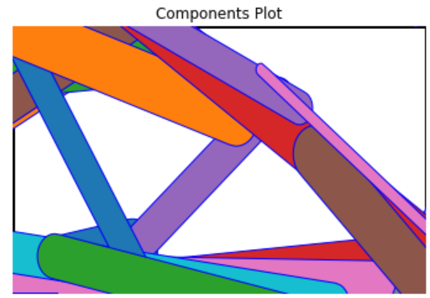
Figure 4.1 shows both the density plot and the components plot for a easy side by side comparison.

The density plot shows in black the elements with full material ($x_e = 1$), in white the void ($x_e = 0$) and in different shades of grey the intermediate densities that correspond to the boundaries of the components. These intermediate densities were penalized by the SIMP method (Section 2.2.2).

The components plot shows the optimized components, that are defined in polar coordinates by: the positions of the centers x and y, the heights (L), the widths (h) and the angles (θ).



(a) Density Plot



(b) Components Plot

Figure 4.1: Comparison between density and components plots.

The evolution of the objective function c, is plotted in Figure 4.2. It confirms, as expected, that the compliance starts with a large value and is constantly minimized, reducing the strain on the structure and improving the stiffness.

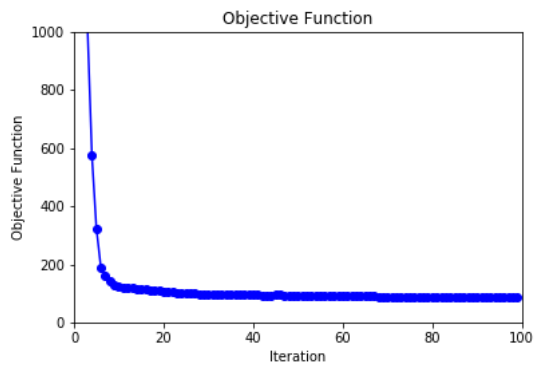


Figure 4.2: Compliance.

The evolution of the volume fraction is plotted in Figure 4.3. It can be seen that the volume fraction starts with a value near 0 and quickly increases until the volume constraint of 50%. When near the volume constraint it varies slightly while trying to minimize the objective function.

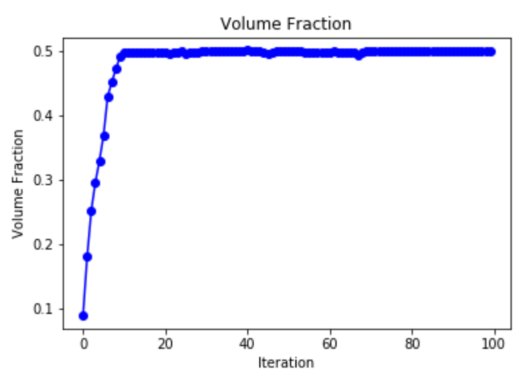


Figure 4.3: Volume fraction.

The value of kktnorm is what dictates if the volume fraction will increase or decrease. In Figure 4.4 it can be seen that when the volume fraction is far from the constraint, kktnorm has a large value and when the volume fraction is already near the 50% volume constraint, the kktnorm has a value near 0.

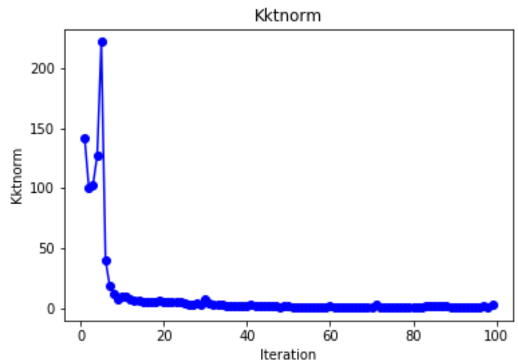


Figure 4.4: Kktnorm.

The less material in a structure, the less stiff the structure is. In Figure 4.5 is plotted the graph of the compliance as a function of the volume fraction and it shows the density plots for 20%, 30%, 40%, 55% and 70% of volume fraction. The figure proves that the higher the volume fraction of material in the structure, the smaller the compliance in the structure is, which means that the structure has bigger stiffness.

The volume fraction is one of the inputs in topology optimization and engineers try to find the perfect equilibrium between lighter and stiffer structures.

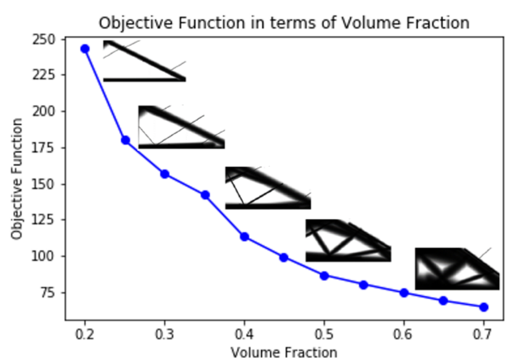
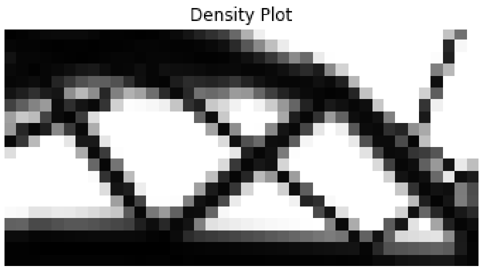


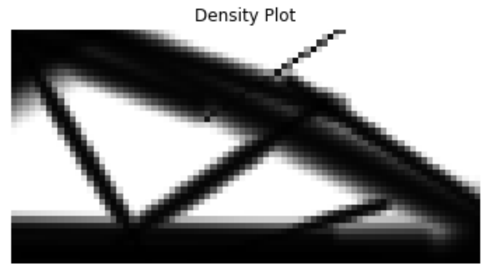
Figure 4.5: Objective function per volume fraction.

Another input on the topology optimization code is the number of elements in x and y. In Figure 4.6 four different but proportional mesh sizes were run. The figure confirms that, as expected, a different mesh size does not change substantially the final result, but a bigger mesh size gives a more accurate result.

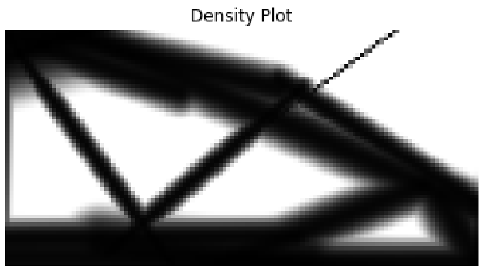
Having 40 x 20 elements in Figure 4.6a is not accurate enough to get well defined components, but having 160 x 80 elements in Figure 4.6d is already a big enough mesh size to be accurate. Engineers often try to find the equilibrium between the mesh accuracy and the computational time.



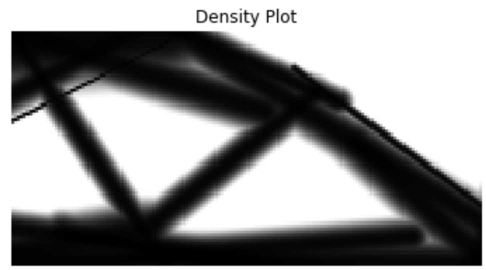
(a) 40 x 20 elements



(b) 80 x 40 elements



(c) 120 x 60 elements



(d) 160 x 80 elements

Figure 4.6: Comparison between mesh sizes.

5 Conclusion and Perspectives

The main achievement for this semester was to develop a SIMP topology optimization implementation in Python to be able to use OpenMDAO next semester.

Next semester it will be added to the code the top rib boundary conditions to optimize a wingbox. As well as two more method approaches: the Moving Node Approach (MNA) by Overveld (2012) and the Geometry Projection (GP) by Norato (2015). These methods are almost implemented but they still need some improvement.

OpenMDAO will be used for the aerostructural coupling, a really important part of the project, to optimize the wingbox having in attention both the structure and aerodynamic forces. For the next semester the main focus will be on OpenMDAO. This platform will be used to create a wingbox and to add the aeroelasticity constraints using J. Mas Colomer's work. Finally all of the wingbox design will be applied to an innovative aircraft, a blended wing body.

Acknowledgements

This work was partly supported by the french « Programme d'Investissements d'avenir » ANR-17-EURE-0005 conducted by ANR.

References

Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., and Sigmund, O. (2011). Efficient topology optimization in matlab using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43(1):1–16.

Gray, J., Moore, K., and Naylor, B. (2010). Openmdao: An open source framework for multidisciplinary analysis and optimization.

Guo, X., Zhang, W., and Zhong, W. (2014). Doing topology optimization explicitly and geometrically—a new moving morphable components based framework.

Hayoung Chung, John T, H. J. S. G. H. A. K. (2019). Topology optimization in openmdao.

Norato, J. (2015). A geometry projection method for the optimal distribution of short fiber reinforcements. *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*.

Overveld, J. (2012). The moving node approach in topology optimization. *Master Thesis, TU Delft, Delft University of Technology*.

Sigmund, O. (2001). A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 21(2):120–127.

Svanberg, K. (1987). The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering*.