Blended Wing Body Wingbox Design with Aeroelasticity Constraints

João C.F. Matos
ISAE-Supaero, 31400 Toulouse, France

Joseph Morlier
ISAE-Supaero, 31400 Toulouse, France

Simone Coniglio
Airbus Operations S.A.S., 31060 Toulouse, France

Abstract

The work presented here is a topology optimization implementation in Python that successfully optimizes a wing rib. The implementation in Python uses the SIMP approach to topology optimization and has four different methods to the boundary penalization: the Generalized Geometry Projection (GGP), the Geometry Projection (GP), the Moving Nodes Approach (MNA) and the Moving Morphable Components (MMC). For the solver it uses its own translated version to python of the Method of Moving Asymptotes (MMA) and for the boundary conditions it has the wing rib and three other test cases: MBB Beam, Short Cantiliever and L-shape.

The following steps that can be done by future ISAE-Supaero students to conclude the project are: designing a wingbox with multiple ribs and by using multidisciplinary optimization add the aerodynamic forces applied on the wing to the designing process. This wingbox design will be applied to 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.

Keywords

Topology Optimization, Wing Rib, Wingbox, Generalized Geometry Projection, Method of Moving Asymptotes

I. Nomenclature

MMA	=	Method of Moving Asymptotes
GGP	=	Generalized Geometry Projection
MMC	=	Moving Morphable Components
MNA	=	Moving Node Approach
GP	=	Geometry Projection
BWB	=	Blended Wing Body

II. Introduction

This project was made to be pursued in the topic of blended wing body wing body design with aeroelasticity constraints. The project had a duration of one year and was 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 multiple wing rib profiles that together constitute a wingbox that takes into account the aeroelasticity and apply it to an innovative design, a blended wing body design (BWB).

A. Objectives

The full long term 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 successfully completed. In this report it will only be covered the first two steps, these two were achieved while the remaining three are left to be continued by future ISAE-Supaero students.

1) Develop a topology optimization implementation in Python.
2) Add wing rib 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 was the need to use a topology optimization code that optimizes a wing rib.

Due to the fact that OpenMDAO is only compatible with Python, the already developed topology optimization implementation could not be used. So a new topology optimization implementation was the main goal for this two semesters.

B. Topology Optimization

The performed 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 [1] 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 1. The objective of this problem is to minimize the compliance.

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

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 input 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 [2]. In this implementation it was only used the Method of Moving Asymptotes (MMA).

C. 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 [1] 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 [3] and by Andreassen et al. [4] present and explain simple implementations with the SIMP method.

D. Moving Morphable Components

In the morphable components method by Guo et al. [5] are used morphable component sas builiding blocks. It is a set method [6], that is positive inside a component, equal to zero on the boundary of a component and negative outside the component.

A solution with this method can be seen in Figure 1. This results was gotten for a Short Cantilever, with a mesh size of 60x30 and a 40% material volume constraint.

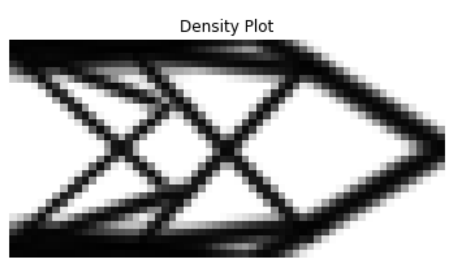


Fig. 1 Example of a MMC approach.

E. Moving Node Approach

The Moving Node Approach was proposed by Overveld [7] in his master thesis. This method has a particularity that there is a weighting parameter in each component, in case two components are overlapped the weights for each components are added and bounded on a maximum of 1. To reduce the computational time in his work, Overveld [7] minimized the number of degrees of freedom.

An example of the MNA the same Short Cantilever, with a mesh size of 60x30 and a 40% volume constraint is displayed in Figure 2.

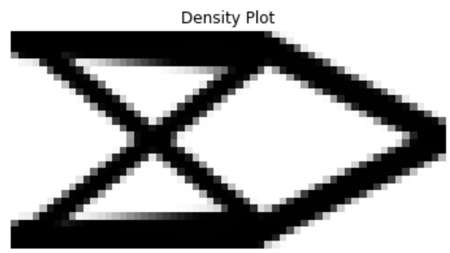


Fig. 2 Example of a MNA approach

F. Geometry Projection

The Geometry Projection method by Norato [8] has the specificity of creating a small circumference of radius r in each element, and computing the density of the correspondent element as the fraction of material in that circumference.

An example of the Geometry Projection method is showed in Figure 3. This result was calculated for the same parameters as the previous methods, for a Short Cantilever, a mesh size of 60x30 and a volume fraction of 40%.

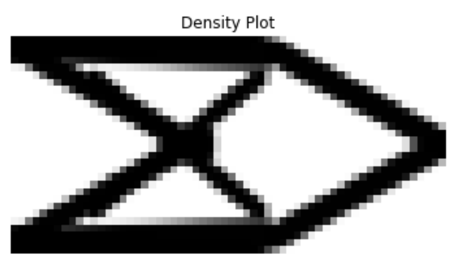


Fig. 3 Example of a GP approach

G. Generalized Geometry Projection

When trying to join the pacrticularities of each of these three previous methods together, Coniglio [9] came up with this new approach, the generalized geometry projection. It uses the Norato's bars (rounded components). This approach can retrieve each of the previous mentioned approaches and these approaches can be replicated as a specific case of the Generalized Geometry Projection.

Figure 4 shows the GGP replicating the MMC method for the same Short Cantilever, mesh size of 60x30 and the 40% material volume constraint.

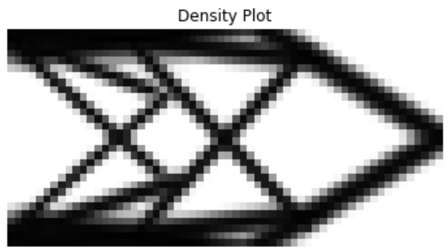


Fig. 4 GGP

The rest of this paper is structured as follows: Section III explains the methodology, section IV presents the Results from the Python implementation and finally section V presents the conclusions and explains the next steps to conclude the project.

III. Methods

The topology optimization implementation developed in Python uses the SIMP approach (Section II.C). Four SIMP approach methods were implemented on the code: the Moving Morphable Components (MMC) [5], the Moving Node Approach (MNA) [7], the Geometry Projection (GP) [8] and the Generalized Geometry Projection (GGP) [9]. 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 2.

$$\begin{cases} min_{\{x\}} c = U^T K U = \sum_{e=1}^{N} (x_e)^p u_e^T k_0 u_e \\ \frac{V(x)}{V_0} \leq f \\ 0 \leq x_{min} \leq x \leq 1 \end{cases} \tag{2}$$

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 [2]. This solver was translated into Python.

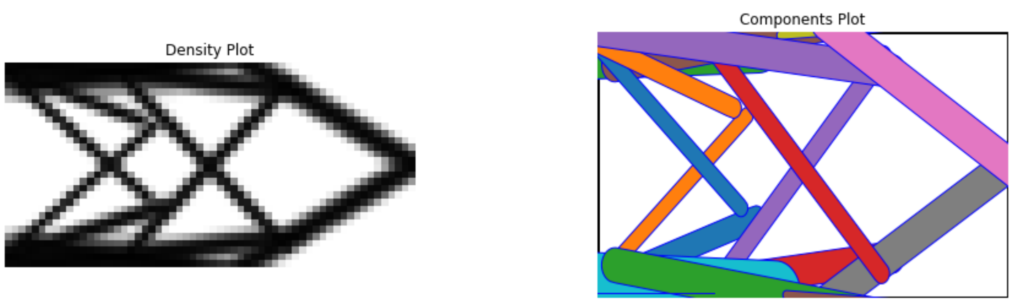
IV. Results and Analysis

When running the topology optimization python code, the program plots the graphs with the results with a rate of $plot\_rate = 10$ iterations. It always plots four different plots: the density plot (Figure 5a), the components plot (Figure 5b), the compliance (Figure 6) and the volume fraction (Figure 7). The following graphs show the GGP results for a mesh of 60 x 30, a Short Cantilever beam and a volume fraction of 40%.

Figure 5 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 voids ($x_e = 0$) 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 II.C) with a parameter $p = 3$.

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



(a) Density Plot (b) Components Plot

Fig. 5 Comparison between density and components plots.

The evolution of the objective function c with the iterations, is plotted in Figure 6. 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.

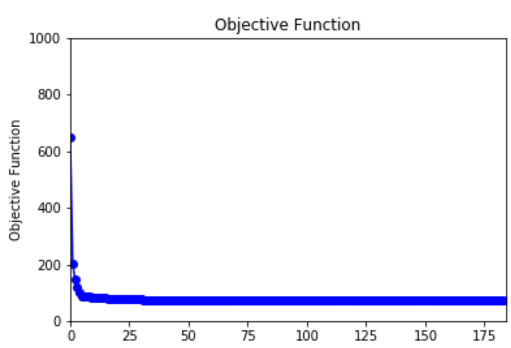


Fig. 6 Compliance.

The evolution of the volume fraction is plotted in Figure 7. It can be seen that the volume fraction starts with a value near 0 and quickly increases until the volume constraint of 40%. Where it starts varying slightly while still minimizing the objective function.

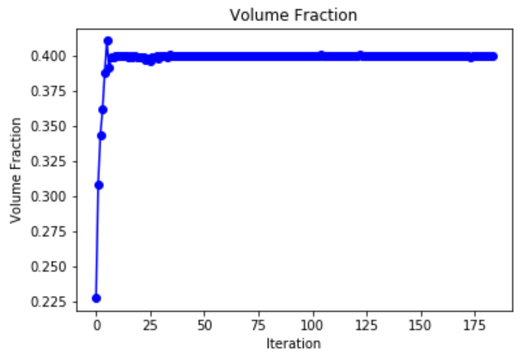


Fig. 7 Volume fraction.

The value of kktnorm is what dictates if the volume fraction will increase or decrease. In Figure 8 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 40% volume constraint, the kktnorm approaches values near 0.

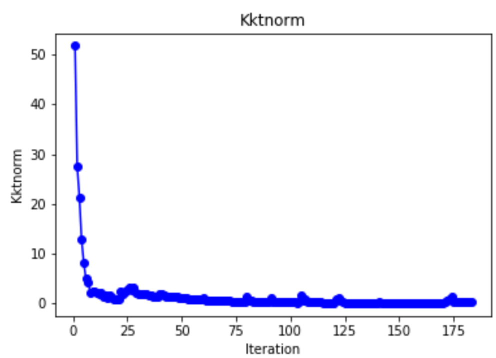


Fig. 8 Kktnorm.

The less material in a structure, the less stiff the structure can be. In Figure 9 is plotted the graph of the minimized compliance as a function of the volume fraction constraint placed 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 it is the job of the engineers to find the perfect equilibrium between lighter while still stiff structures.

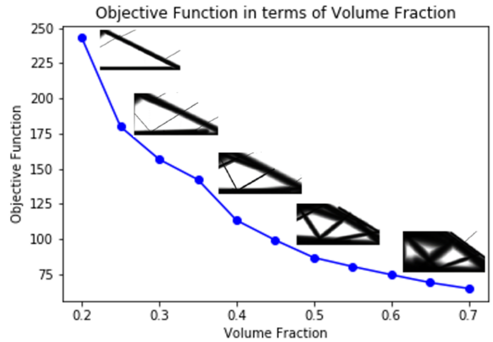
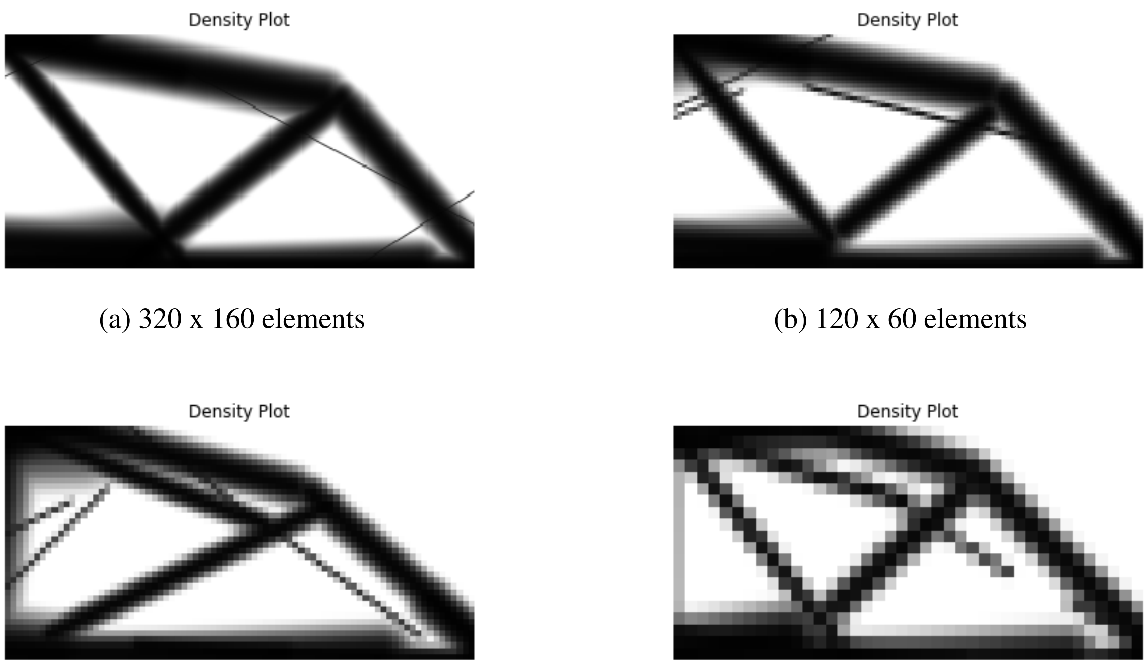


Fig. 9 Objective function per volume fraction.

Another input on the topology optimization code is the number of elements in x and y. In Figure 10 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 and better defined result.

Having 40 x 20 elements in Figure 10d is not accurate enough to get well defined components, but having 320 x 160 elements in Figure 10a it already shows very well defined components. Engineers often try to find the equilibrium between the mesh accuracy and the computational time.



(a) 320 x 160 elements

(b) 120 x 60 elements

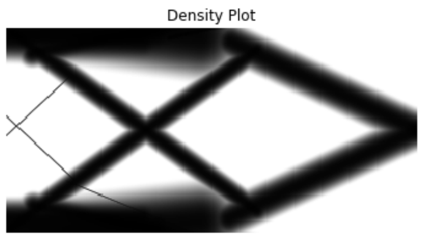
(c) 80 x 40 elements

(d) 40 x 20 elements

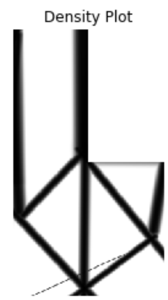
Fig. 10 Comparison between mesh sizes.

In the Python code the four previously mentioned methods are applied, and all give different but reliable results. These are displayed in section II in Figures 1 to 3: the GGP, the MMC, the MNA and the GP for a mesh size of 60x30, the Short Cantilever and a volume constraint of 40%.

Besides the methods, the implementation has test cases, that allow the code to check the validity of its results. These are: the MBB beam, the Short Cantilever, the L shape. After these three test cases validated the implementation, the real main one was implemented, the Top Rib that designs a 2D wing rib.



(a) Short Cantilever, GGP, 40%, 320x160



(b) L-Shape, GGP, 20%, 160x320

Fig. 11 Short Cantilever and L-Shape test cases

The L shape is different than the previous test cases because it has some domain areas that are set to be void. This was a good test to check before introducing the wing rib.

The Top Rib simulates a wing rib. With forces applied all over the boundaries, and with multiple areas set as void. This result is presented in Figure 12 for the GGP method, a mesh size of 160x40 and a volume constraint of 40%.

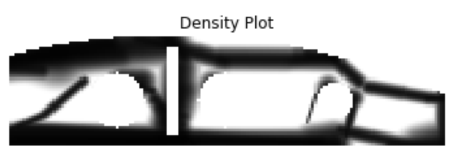


Fig. 12 Top Rib with GGP

The GGP python implementation code showed a limit in the computational speed, when compared to similar implementations in MATLAB.

Besides the main GGP implementation code, a separate and purely MNA Top Rib code was developed.

This code is much faster due to being specifically made for MNA Top Rib and is recommended to be used for the continuation of the project when proceeding with the project in multidisciplinary optimization. An example of a result for this code is presented in Figure 13 for a mesh size of 160x40 and a volume fraction of 40%

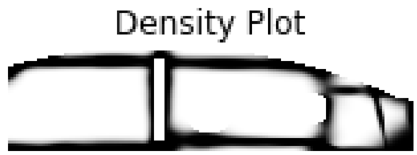


Fig. 13 Alternative MNA Top Rib code

V. Conclusion and Perspectives

The main achievement for these two semesters was to develop a SIMP topology optimization implementation in Python of a wing rib profile. This goal was successfully accomplished with the development of a Python code that has the GGP, MMC, MNA and GP methods implemented and it the wing rib boundary conditions. Because this Python script is very slow and should not be used with in the multidisciplinary optimization, an alternative more focused to a wing rib with MNA has also been developed. This one is able to be more efficient in the calculations because it was made specifically for the MNA method. Both this codes use a MMA solver that was translated into Python.

For future students that wish to follow the project and create a wingbox optimization, they are recommended to use OpenMDAO for the aerostructural coupling, to optimize the wingbox having in attention both the structure and aerodynamic forces. 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 can 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

[1] Hayoung Chung, H. J. S. G. H. A. K., John T, "Topology Optimization in OpenMDAO," 2019.

[2] Svanberg, K., "The method of moving asymptotes—a new method for structural optimization," *International Journal for Numerical Methods in Engineering*, 1987.

[3] Sigmund, O., "A 99 line topology optimization code written in Matlab," *Structural and Multidisciplinary Optimization*, Vol. 21, No. 2, 2001, pp. 120–127. https://doi.org/10.1007/s001580050176, URL https://doi.org/10.1007/s001580050176.

[4] Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., and Sigmund, O., "Efficient topology optimization in MATLAB using 88 lines of code," *Structural and Multidisciplinary Optimization*, Vol. 43, No. 1, 2011, pp. 1–16. https://doi.org/10.1007/s00158-010-0594-7, URL https://doi.org/10.1007/s00158-010-0594-7.

[5] Guo, X., Zhang, W., and Zhong, W., "Doing Topology Optimization Explicitly and Geometrically—A New Moving Morphable Components Based Framework," 2014.

[6] Allaire, J. F. T. A., G., "Structural optimization using sensitivity analysis and a level-set method." *Journal of computational physics*, 2004.

[7] Overveld, J., "The Moving Node Approach in Topology Optimization," *Master Thesis, TU Delft, Delft University of Technology*, 2012.

[8] Norato, J., "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*, 2015.

[9] Coniglio, M. J. G. C. e. a., S., "Generalized Geometry Projection: A Unified Approach for Geometric Feature Based Topology Optimization." *Arch Computat Methods Eng*, 2019.

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
31/03/2020

Signature
João Matos