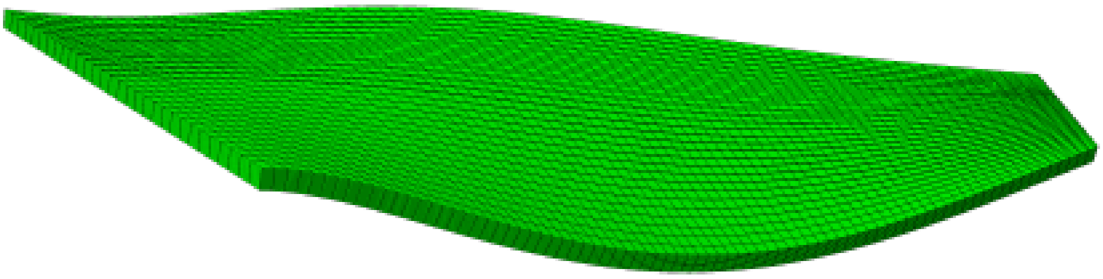


Design Optimization of Dynamic Scaled Wing Models Using Multi-Material Topology Optimization

Alessandro Brogliato

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisors: Dr. Abdolrasoul Sohouli
Prof. Afzal Suleman

November 2021

Acknowledgments

I would like to acknowledge and give my warmest thanks to doctor Abdolrassoul Sohouli, his patience and disponibility during this work motivated me to proceed despite all the difficulties that arised during this circuitous period.

I would like to thank professor Afzal Suleman for giving me the opportunity to participate to the Advanced Composites Group and for being always present.

Thank you to my family for always supporting me, both in making difficult decisions and in moments of despair. Thanks to mum for understanding me without me even saying a word, thanks to my dad for showing me that I need to be strong no matter what and thanks to my brother because you are able to continuously surprise me with your perseverance and your calm.

Thanks to my ”triumviri” Alessandro, Francesco and Giacomo, because friendship goes over distances and time and you have always been there.

Thanks to my Portuguese family, Miki and Renata.

Thanks to Pati.

And finally thanks to the sea, the element that calms me down and keeps me going.

Abstract

The dynamic behavior of an aircraft can be investigated through computation or experiments. To validate their designs, engineers need to perform experiments that are usually expensive and time-consuming. In order to improve this process, topology optimization can be used to produce scaled models with the same dynamic behavior of the full-scale model but that requires smaller facilities to be tested and less time to be produced thanks to the potentialities of additive manufacturing. The goal of this work is to produce a scaled model that has the same dynamic behavior as the full-scale model. The method used to obtain this result is the multi-material topology optimization that the material properties are defined using the Solid Isotropic Material with Penalization method. The Modal Assurance Criterion is selected as the optimization constraint to track and synthesize modes between the full scale and the scaled designs. Eigenvector derivatives must be calculated and two different methods were used in order to calculate the sensitivities.

Keywords

Dynamic Scaled Model, Eigenvectors constraints, Additive manufacturing, Dailey's method, Adjoint method.

Resumo

O comportamento dinâmico de uma aeronave pode ser investigado através de simulações computacionais ou testes experimentais. Para validar os seus projetos, os engenheiros precisam de realizar testes que geralmente são dispendiosos e demorados. Para melhorar este processo, a otimização da topologia pode ser usada para produzir modelos à escala com o mesmo comportamento dinâmico do modelo à escala real, mas que requerem menores instalações para serem testados e menos tempo para serem produzidos graças às potencialidades da manufatura aditiva. O objetivo deste trabalho é produzir um modelo à escala que tenha o mesmo comportamento dinâmico do modelo à escala real. O método utilizado para obter estes resultados é a otimização topológica multi-material, em que as propriedades são definidas usando o método do Material Isotrópico Sólido com Penalização. O Critério de Garantia Modal é incluído como uma restrição na otimização para seguir e sintetizar os modos entre os modelos à escala e real. As derivadas dos vetores próprios são calculadas utilizando dois métodos distintos, para obter as suas sensibilidades.

Palavras Chave

Modelo em escala dinâmica, Restrições de vetores próprios, Manifatura aditiva, Dailey's method, Adjoint method.

Contents

List of Figures

List of Tables

Acronyms

AESO	Additive Evolutionary Structural Optimization
AM	Additive Manufacturing
BESO	Bi-directional Evolutionary Structural Optimization
DLP	Digital Light Processing
FEA	Finite Element Analysis
ESO	Evolutionary Structural Optimization
FDM	Fused Deposition Modeling
MAC	Modal Assurance Criteria
MMTO	Multi Material Topology Optimization
NLP	Non Linear Programming
RAMP	Rational Approximation of Material Properties
RP	Rapid Prototyping
SAO	Sequential Approximate Optimization
SIMP	Solid Isotropic Material with Penalization
SLA	Stereolitography

SLM	Selective Laser Melting
TO	Topology Optimization
IPOPT	Interior Point Optimization
PTO	Proportional Topology Optimization

1

Introduction

Contents

The engineer bases his existence on the concept of optimization. Usually there are constraints that limit the amount of time or money that can be involved in the process and the task of the engineer is to obtain the best results from the given amount of resources.

The figure of the engineer changed constantly during the history but received a drastic mutation after the invention of the modern computer. The machine conceptualized by Alan Turing in 1936, in fact, while being initially limited to simple operations, allowed the engineer to perform calculation that would have required unreasonable amounts of time otherwise.

In this way the engineer could avoid drastically the trial and error process allowing for a great saving in time and resources.

1.1 Topology optimization

In this scenario of continuous technological improvement the concept of Topology Optimization introduced during the 1960s was applied by Bendsøe and Kikuchi [1] in 1988 at the beginning as a static problem where the optimization targets was the displacement field.

While engineers can usually propose new designs of simple parts using the experience, the problem becomes more complex when the number of loads and constraints increases, this is the point where Topology Optimization (TO) becomes beneficial. TO is the process of defining a distribution of material inside the design domain in order to maximize a given objective function. For example, if we consider an aircraft, the wings have to sustain variable force fields and therefore the optimization can involve not only the maximization of the load tolerable by the structure but also the maximization of the dynamic properties of the structure defined by the eigenproblem:

$$(K - \omega^2 M) = 0 \tag{1.1}$$

where K is the stiffness matrix, ω is the natural frequency and M is the mass matrix. The solution to this problem consists in a number of values called eigenfrequencies that connote the resonant frequencies of the structure. When the structure vibrates at a certain frequency, called natural frequency, the vibration amplitude tends to diverge causing failure. The eigenvectors are associated to these eigenfrequencies and define the displacement field of the structure at that defined eigenfrequency.

Using Multi Material Topology Optimization (MMTO) [2] it is possible to modify the eigenfrequencies and the eigenvectors associated to a scaled structure in order to match the full scale structure in a process called dynamic scaling.

The difference between MMTO and TO is that during the optimization process are not considered only two materials one of which is void but more than two allowing for a higher precision in the results. One of the most used methods in TO is the Solid Isotropic Material with Penalization (SIMP) and is the one used in this optimization process [3].

During the process of optimization Modal Assurance Criteria (MAC) can be used to monitor the evolution of the eigenmode of the optimized structure in comparison to the objective structure. An inequality constraint based on MAC can be implemented inside the optimization environment Interior Point Optimization (IPOPT) in order to guide the optimization to have similar mode shapes between the two structures [4].

In order to calculate the derivatives of MAC, the derivatives of the eigenvector have to be calculated [5]. This process can be performed with different algorithms such as Nelson's, Dailey's or Friswell's [6] [7] [8]. The disadvantage of the first method is that it is not usable in the not remote case of multiple eigenvalues, this is why the Dailey's Method was firstly applied to this research without satisfying results. For this reason a different method was used in order to calculate the MAC derivatives, this so called Adjoint Method was presented in a paper called ”Structural design for desired eigenfrequencies and mode shapes using topology optimization” [5] in 2013 and provides a fast and efficient way to obtain MAC derivatives. One of the main advantages of the method is that it performs the calculations using global matrices and therefore linearly solves a system in order to obtain the solution resulting faster than the previously mentioned Dailey's method. The biggest disadvantage is that the method does not consider multiple eigenvalues.

1.2 Goal of the optimization

The optimization process adopted in this work aims to generate a scaled model of a generic structure that presents similar characteristics to a compared full scale structure. In particular the structure in exam is a wing-box. The objective function of the optimization will be the minimization of the difference between the eigenfrequencies of both models and the inequality constraints will based on the MAC. This approach will guide the optimization to obtain, as mentioned before, matching natural frequencies and eigenmodes.

1.3 Additive manufacturing

Once the scaled model is dynamically similar to the full scaled model and the optimization reached its goal, a prototype can be manufactured and tested inside a controlled environment.

An efficient and fast way to produce the optimized structure can be additive manufacturing. This field undergone a great development in the last years and has proven to be one of the best alternatives to produce a prototype [9]. For this reason Additive Manufacturing (AM) can be also mentioned as fast prototyping.

Modern machines can produce pieces using multiple materials such as titanium, brass, aluminum alloys and polymers. Thanks to the increased precision that these machines can achieve, lattice structures can be considered increasing the potentialities and the flexibility of the process [10].

1.4 Structure of the document

The main goal of this work is to introduce a constraint to the optimization in order to obtain matching eigenmodes between the full scale and the scaled models. In order to do so the main target was to implement a constraint function and to calculate the sensitivity of the eigenvectors using the Adjoint Method and to introduce them in the Matlab code.

The document is going to introduce initially the problem and the optimization environment used. After that, the calculation of the aforementioned sensitivities is presented with the inclusion of the calculation of the MAC derivatives.

The chapters of results presents a series of optimizations performed with different material parameters and the best results are presented more in details.

The document ends with a final chapter where the problems encountered are presented and possible solutions are proposed for future developments of the work.

2

Topology Optimization

Contents

TO is the mathematical process that defines where to place material inside the design space in order to maximize the performance of the structure [11]. The concept was firstly introduced by Bendsøe and Kikuchi in 1988 [1] and since then it revolutionized the approach of design in several fields such as fluid dynamics, optics, acoustics and many more.

The method is based on multiple analysis, one per every iteration, of the design and the update of the topological distribution of the material in the domain under the guide of the gradient computation. [11]

The development of this technology allowed engineers to optimize the structure without the objective of maximizing some parameters such as stiffness or the first natural frequency but to obtain structures with similar dynamic behaviour. This is the case analyzed in the following pages where the dynamic properties of a scaled wing-box are going to be optimized in order to match the properties of the full scaled wing-box.

Before entering in detail with the process adopted for the optimization, a brief description of the most common methods used is presented.

2.1 Solution Methods

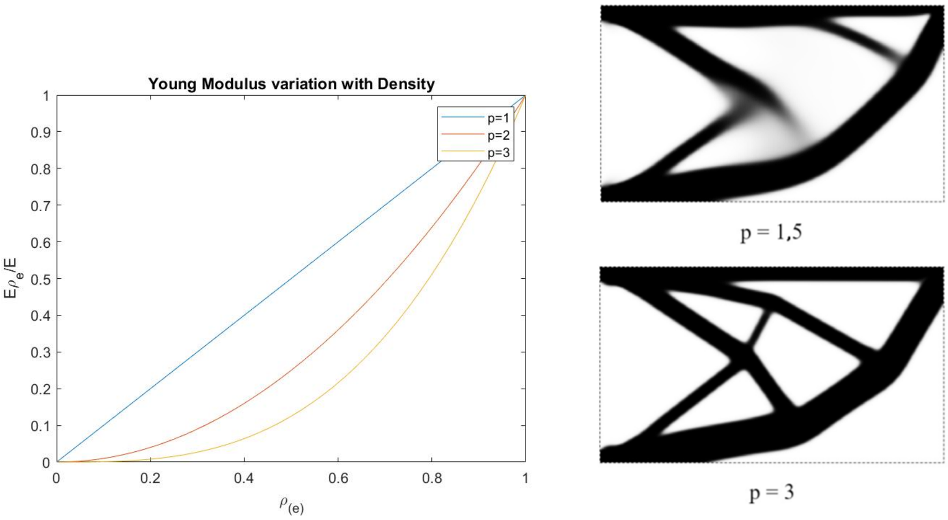
There are multiple solving methods to be used in topology optimization. Some of them have been used since decades and have proved their solidity such as SIMP and Evolutionary Structural Optimization (ESO). Other methods as Proportional Topology Optimization (PTO) have been recently proposed and can take advantage of the increased computational power available in order to perform more accurate calculation. Some of these methods are going to be introduced in the following paragraphs.

2.1.1 SIMP and RAMP

Solid Isotropic Method with Penalization has been proposed by Bendsøe and Kikuchi [1] in 1989 and then by Rozvany and Zhou in 1991 [3]. The method was proposed in order to prevent the formation of grey areas in the optimized structure. Gray areas are particularly problematic because they represent a material that was not expected at the beginning of the optimization or simply does not exist. The stiffness is defined by the following formula:

$$E(\rho_e) = \rho_e^p E_0 \tag{2.1}$$

E defines the Young Modulus of every element in the discretization and ρ is the density relative to the element. This definition indicates that if p=1, E/E_0 varies linearly between 0 and 1 while if $p > 1$, E/E_0 assumes an exponential behaviour prioritizing external values and therefore avoiding grey areas. A representation of the variation of the Young Modulus with the penalty factor and its impact on the results is presented is figure 2.1.



(a) Young modulus variation with p

(b) Impact of p factor in the result [12]

Figure 2.1: Penalization factor

A variation of the SIMP method is the Rational Approximation of Material Properties (RAMP) method. In this case the Young modulus is defined by:

$$E(\rho_i) = w(\rho_i)E_0 \quad where \quad w(\rho_i) = \frac{\rho_i}{1 + p(1 - \rho_i)} \tag{2.2}$$

Thanks to this modification the RAMP [13] method has the gradient different from zero when ρ_i=0.

2.1.2 Evolutionary Structural Optimization

The ESO has been proposed in 1992 by Xie and Steven [14] and is based on the assumption that some areas of the structure are subjected to very low stresses and therefore these areas can be removed iteratively until the structure reaches the optimal solution. The process of selection is quite simple: if the Von Mises stress of the cell is lower than the maximum stress tolerated by the material multiplied by a factor called rejection ratio (RR) the material present in the cell can be removed. The method proved to be effective in particular scenarios but had one main issue: once the material has been removed it can't be reintroduced.

Following this results a different method was proposed, the Additive Evolutionary Structural Optimization (AESO) method considers a kernel structure that is overly stressed and applies material in the zones that present a concentration of stresses [15].
In 1999 Bi-directional Evolutionary Structural Optimization (BESO) was introduced. The method was developed in order to solve two problems related the the previous evolutionary method: the uniqueness of the solution and the addition of material once removed [16]. The method can start from a kernel structure or an oversized structure and remove or add material according to the stress distribution of the structure.
A representation of the evolution is presented in figure 2.2:

- In subfigure **a** the kernel structure is presented.
- In subfigure **b** material is added by the algorithm in order to reduce de stresses
- In subfigure **c** it is possible to notice that the iterations added material to the external surface while removing material from the center.
- In subfigure **d** the behaviour expressed in the previous point is accentuated.

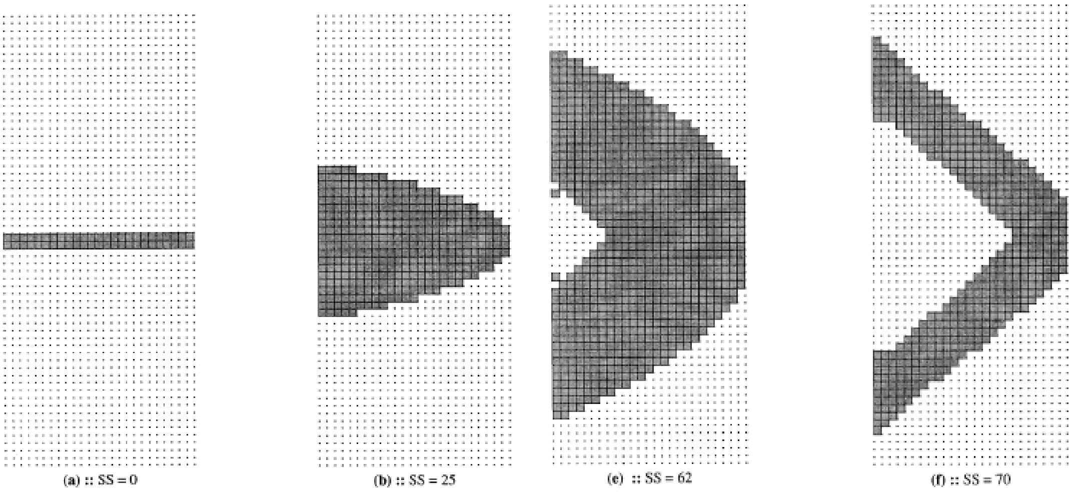


Figure 2.2: BESO [16]

2.1.3 Proportional Topology Optimization

PTO is a non sensitivity method and therefore it doesn't require the analytical derivation of the sensitivities of the structure. It was at first proposed by Biyikli and To in 2014 [17] and later improved by Wang, Cheng and Du in 2020 [18]. In these methods the density variable is updated with at every iteration and the target of the optimization can be the minimization of compliance or the minimization of the maximum stress. The SIMP method is implemented inside PTO and it possesses a great simplicity and a good grade of accuracy for the given simplicity while avoiding the calculation of the derivatives relative to the structure.

2.2 Checkerboard patterns

A concerning issue in topology optimization is the formation of checkerboard patterns, and it is presented in figure 2.3. These patterns are areas in the design domain where the optimizer converged to a physically unfeasible solution where cells with materials alternate with cells without material.

Initially, it was believed that this solution was representing the optimal micro-structure but this thesis was denied and the formation of checkerboard patterns were due to bad numerical modelling [19].

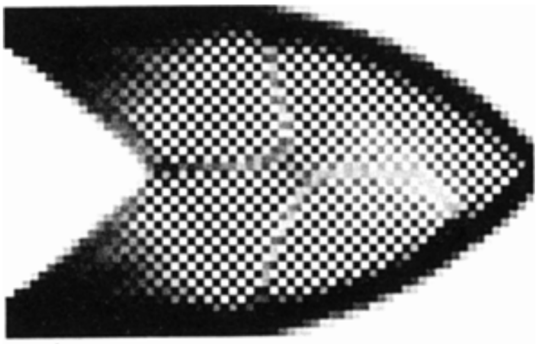


Figure 2.3: Checkerboard pattern [20]

In order to avoid the formation of this problem some strategies can be adopted: at first higher-order finite element can be used but this solution cannot be applied to all methods. Smoothing is an option to be avoided because it adopts image processing in order to remove the checkerboard but doesn't address the problem that generated the pattern. Another solution could be to use filtering techniques [21], those techniques try to smooth the design sensitivity of the element similarly to the techniques used in the imaging processing. The element is therefore influenced by its closest neighbors and the checkerboard is less likely to occur; the downside of this technique is that adds computational cost.

2.2.1 Mesh dependency

The mesh-dependency problem is well illustrated in figure 2.4, where the optimized structure is highly dependent on the size of the mesh adopted. This means that the structure present in figure **c**, that is obtained using a finer mesh, contains more substructures than figure **d** while it should simply be a smoother version of it. This can be evaluated just counting the number of beams that structure c and d present.

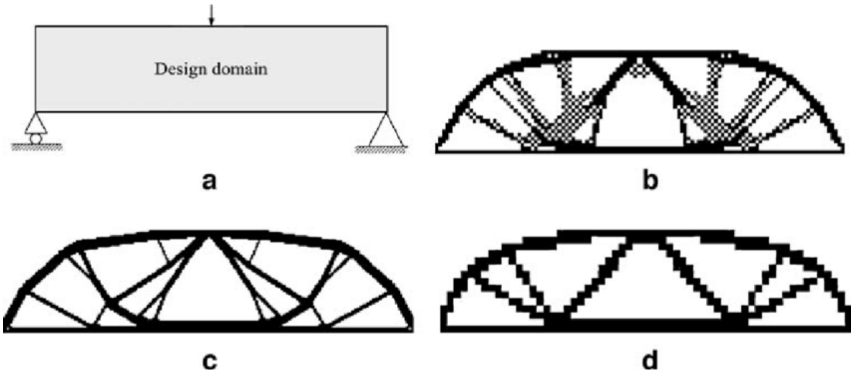


Figure 2.4: Mesh dependency [22]

The problem illustrated in figure 2.4 is dependent on the non existence of the solution, but the problem of mesh-dependency can present itself also in situation where there are multiple optimal solutions, such as a bar under uni-axial tension.

2.2.2 Density Filter

When bad results such as checkerboard patterns appears, a filter can be applied to the structure in order to improve the results. In this work a density filter is applied to the structure [23], the formulation of the filter is presented by equation 2.3:

$$\vec{f_e} = \frac{\sum_{i \in N} H_{ei} f_i}{\sum_{i \in N} H_{ei}} \tag{2.3}$$

Where $H_{ei} = max(0, r_m - \Delta(e,i))$ and N is the neighborhood or, in other words, the set of elements that are located at a distance lower than r_m. As it is presented in figure 2.5 the filter applies only to the elements that are interested by the filter radius and it averages the value of the neighborhood in order to avoid the checkerboard formation.

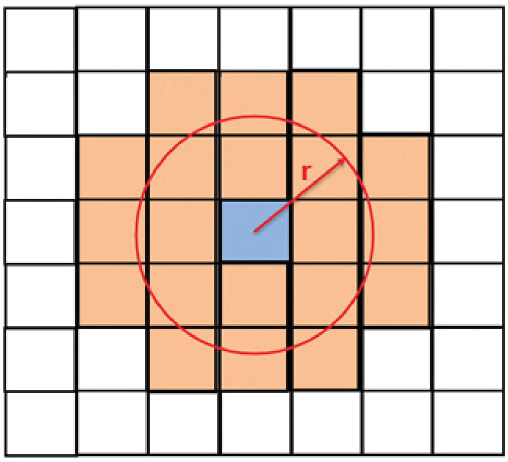


Figure 2.5: Filter radius

2.3 Discrete material optimization

Discrete material optimization [24] is a particular type of topology optimization where the algorithm doesn't have to choose between solid and void but between a set of intermediate densities. The set of available materials takes the name of candidate materials and the total number of design variables becomes equal to the number of elements multiplied by the number of materials. In order to adapt the SIMP to the MMTO the constitutive matrix E_p and the density ρ_p are defined as follows, where NW is the number of material selected for the optimization including void:

$$\begin{cases} E_p = \sum_{j=1}^{NW} \omega_j^E E_j \\ \rho_p = \sum_{j=1}^{NW} \omega_j^\rho \rho_j \end{cases} \tag{2.4}$$

w_j^E and w_j^ρ are the weight functions and can be described by the following equations:

$$\begin{cases} w_j^E(x_p) = \frac{x_p}{1+pen(-x_p+1)} \\ w_j^\rho = x_p^{pen} \end{cases} \tag{2.5}$$

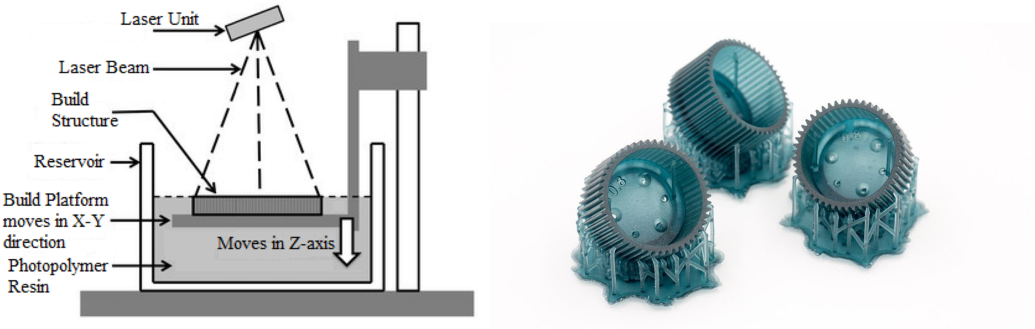
2.4 Additive Manufacturing

AM was originally referred as Rapid Prototyping (RP) for its ability of rapidly creating a prototype. The research on this topic was conducted in the 1970s but the process was patented by Charles Hull in 1986 as a process called Stereolitography (SLA), also known as optical fabrication [25]. The term AM describes the principle of operation of the method where sequential layers of the material are additively stratified one on the other in order to produce the component.
AM is widely used in the industry as a prototyping tool but lately it spread also in the hobbyist world thanks to the reduction in price of SLA and Fused Deposition Modeling (FDM) printers.
In order to manufacture a piece it needs to be designed inside a CAD environment. The 3D model is afterwards sliced with a standalone software or with a CAD integrated tool and then uploaded to the printer.

2.4.1 Stereolitograpy

SLA was the first AM method to be developed and uses a focused ultraviolet laser beam in order to solidify a resin that is a thermoset polymer. The laser might be produced by a single source and then directed to the liquid by a moving mirror, can be produced by a matrix of lasers that directly draw the section (improving the speed of the process) or can be produced by a Digital Light Processing (DLP) projector. A schematic representation of an SLA printer is presented in figure 2.6(a).

Even though the technology was introduced 4 decades ago it is still used nowadays because it is relatively fast, has a good resolution and can produce objects with good mechanical properties such as the gears represented in figure 2.6(b). A big disadvantage related to it is the necessity to perform a finishing wash using isopropyl alcohol in order to remove any trace of



(a) SLA scheme [26]

(b) Gears produced with SLA process and high tensile strength resin [27] uncured liquid from the surface and to perform a curing cycle of the finished part and it is important to remark that SLA also produces a considerable amount of material wastes differently from other printing techniques. But recent studies showed the possibility of printing multi-material micro-structures using the SLA process [28].

2.4.2 Fused Deposition Modeling

FDM, also called fused filament fabrication, was patented in 1989 by Stratasys [29] but gained increased popularity after 2009 when the patent expired and the technology could spread.
The printer has one or multiple nozzle that can move on 3 axis and deposit on a printing bed the melted material that hardens after the deposition as represented in figure 2.6.
Most of the time the material is a thermoplastic polymer and therefore the nozzle has to be heated. Thanks to the relative simplicity of the printer its costs are quite low compared to other AM technologies and the wide spread of it also in the hobbystic world generated a wide gamma of usable materials that include recycled, edibles, biological and ceramic filaments.

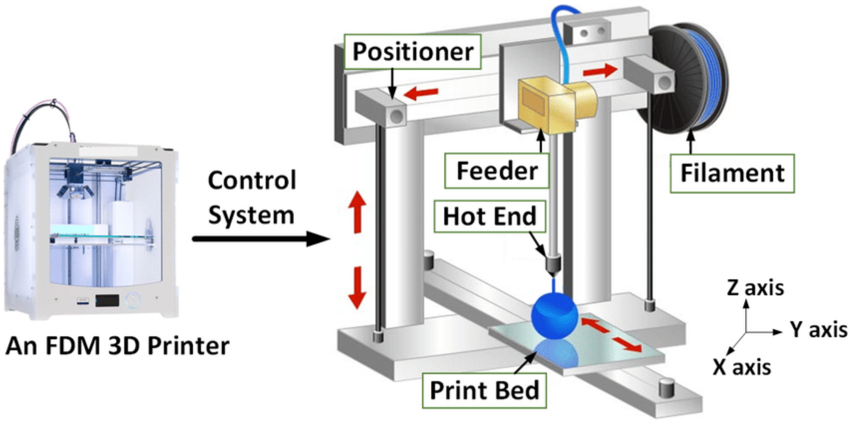


Figure 2.6: FDM scheme [30]

The parts produced with FDM are inexpensive and fast to produce, but according to the nozzle used, they can present quite visible layers that undermine the visual quality and the mechanical properties. Those defects can be reduced using smoothing techniques that require time, chemicals and change the final thickness of the part.
FDM technology can be used to produce big parts out of concrete for architectural purposes [31] and even houses [32].

2.4.3 Selective Laser Melting

Selective Laser Melting (SLM), also known as powder bed fusion or direct metal laser melting, is an increasingly used method for 3D printing. One of the limiting factor, in its utilization, is the high initial and operational cost but it allows to produce complex parts using metal alloys [33]. The produced part has to be smoothed because it presents an high roughness and therefore is not suitable for painting or visual application.

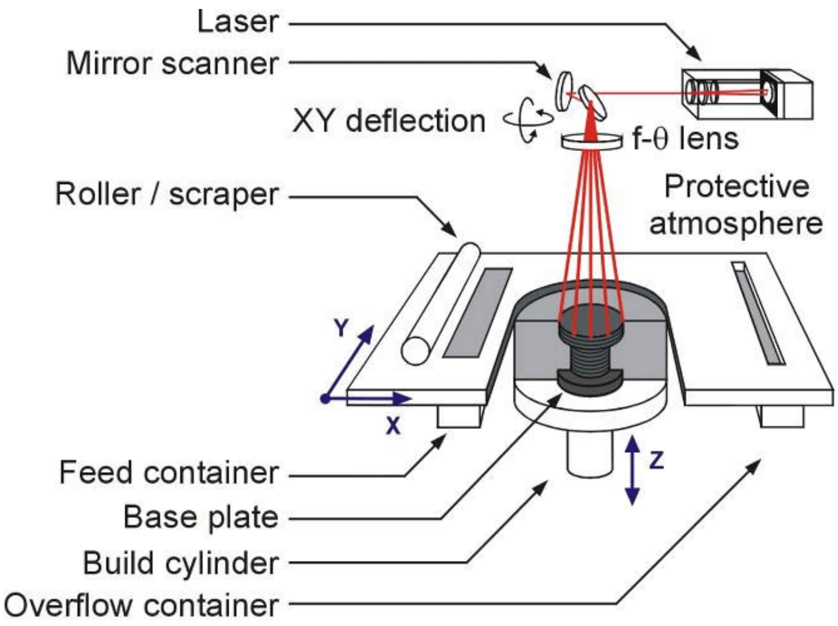


Figure 2.7: SLM scheme [34]

SLM machines are usually big compared to other AM machines, the printing chamber needs to be inert and usually is filled with Argon gas. The printing bed can move vertically and descend of one *layer height* once the layer is completed. Simultaneously another piston that acts as a reservoir provides the powder to the re-coater cylinder that distributes a new layer of powder on the printing bed. As presented in figure 2.7, a laser beam is deflected and focused by a series of mirrors and lenses in order to melt the powder and to generate the final product.
The main advantage of SLM is the possibility to print Aluminum, Titanium, Nickel, Cobalt, Copper and Iron based metal alloys that have similar strength characteristics compared to the same material produced with different techniques. Unfortunately the micro-fluid dynamic characteristics of SLM causes the formation of micro-cavities that decrease the fatigue properties of the final part, especially in iron based parts [35].

2.4.4 Lattice Structures

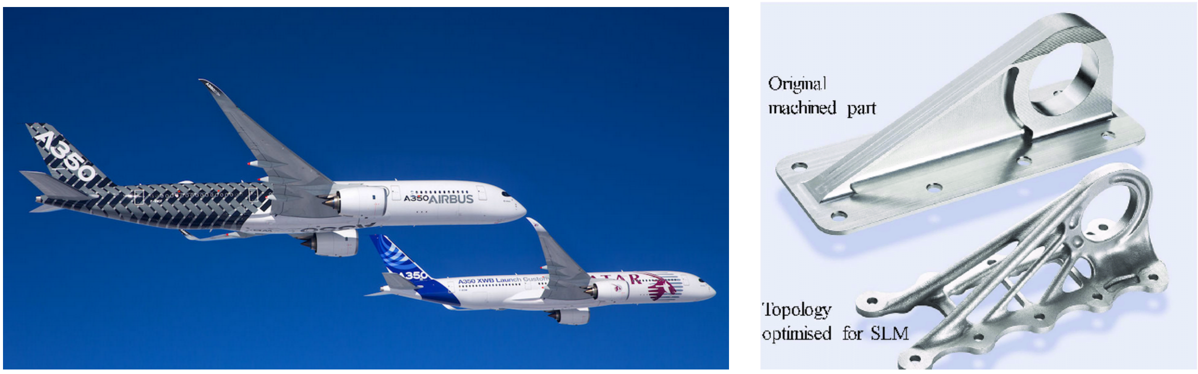
The design of lattice structures using lattice cells derives from the necessity of distribute the material in a more efficient way compared to normal homogeneous materials. This technology provides material with good energy absorption and good thermal and acoustic insulation [36] [37]. In order to build lattice structures the 3D printer, usually using SLM technology [38], produces three-dimensional structures on a micro-scale that are interconnected along their nodes or edges. Usually the cell properties and a microscopical scale are anisotropic but at the component scale those properties results homogenized.

The production of this structures is particularly interesting in the subject of this thesis. Especially considering the new development of lattice structures based on polymers [39], and therefore printed with SLA techniques, that are more suitable for the manufacturing of the core of the wing-box allowing the designer to produce it using intermediate densities that often appear during the optimization process.

2.5 Applications

Started on the first half of the 1980s AM hasn't been extensively used by the industry until the end of the 2010s. The main factors that limited the development of the technology were the patents, most of them are now expired, the cost of the printers, the long production time and the low quality of the prints [40].
While some factors, like the low production ratio, are still a limiting factor for the large scale production, the big design freedom and the low prototyping costs are increasing the popularity of this solution.
Three example of 3D printed parts will be presented to illustrate the state of art of the technology, the examples are drawn from three different industries: aerospace, space and automotive.

2.5.0.A Airbus A350 XWB door hinges



(a) Airbus A350 XWB

(b) Comparison between a CNC machined wing bracket and the topologically optimized used by the A350 [41]

The Airbus A350 XWB presented in figure 2.8(a) is built using over 1000 3D printed parts according to Stratasys, the producer of the printers used by the aircraft manufacturer [42]. This technology allowed to speed up the development process but is still used to produce parts currently utilized on the aircraft because of the unique freedom of design that the technology permits; an example of this is the wing bracket presented in the image 2.8(b) that is produced with SLM technology and a Titanium alloy. Titanium is not the only 3D printed material used on the A350, Ultem is a polyetherimide that is commonly used thanks to its great thermal and flame retardant characteristics while providing also a good chemical resistance.

2.5.0.B Bugatti brake caliper

Bugatti is an historical car manufacturing brand that always succeeded in producing some of the more iconic, high performance and expensive hypercars in the world. Before being acquired in 2021 by the Croatian hypercar brand *Rimac Automobili*, Bugatti was part of the Volkswagen group and could benefit of the Volkswagen advanced research departments that allowed them to design the brake caliper in exam. The brake caliper presented in figure 2.8(d) is produced using a Ti_6AI_4V alloy and compared to a state of the art 8 pistons caliper made out of Aluminum alloys 2.8(c) allows for a saving in weight of the 40% while maintaining the same braking performance.

The new owner of *Bugatti*, *Rimac Automobili*, produces an electric hypercar that takes advantage of advanced 3D printing in several components of the final car.

(c) Brembo 8POT brake caliper

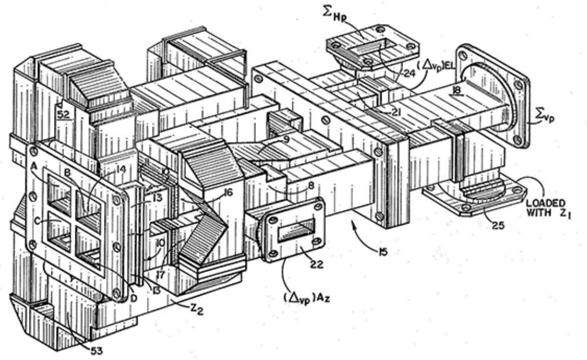
(d) Comparison between a high performance iron cast brake caliper and a SLM 3D printed titanium caliper

2.5.0.C OPTISYS AM antenna

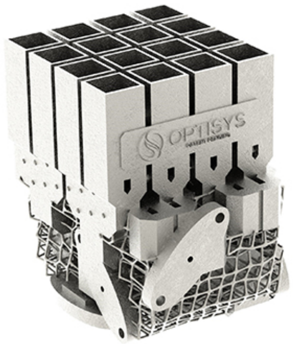
Antennas are critical in satellites communications for conveying information. They usually require large volume and weight 2.8(e) and therefore they can negatively impact the performance of the spacecraft.

The design presented in figure 2.8(f) and produced by the American based company Optisys, allows for a weight saving of 95%, a 80% reduction in production time while still allowing for a reduced production cost around 20%.

Another hidden benefit for 3d manufacturing is that it allows to reduce the part-count, in this case from 100 to 1, therefore reducing the probability of malfunction and the maintenance



(e) Original plan for the antenna assembly



(f) 3D printed titanium optimized antenna

requirements. [43]

3

Optimization

Contents

After describing the basic concepts and various others notions useful to understand the optimization process 2, now it is time to describe the approach used in this research. The optimization problem is going to be described, followed by the integration between the various software available and after that the calculation of the objective function is going to be presented.

3.1 Optimization Problem

The optimization problem studied in this work is to minimize the function f:

$$f = \sum_{i=n}^{n} (1 - \frac{\omega_{scaled}(i)}{\omega_{full}(i)})^2 \tag{3.1}$$

Subject to the constraint *g(i)*:

$$g(i) = 1 + \gamma - MAC(i,i) \quad with \quad \gamma = 10^{-6} \tag{3.2}$$

where ω is defined by equation 1.1 and *MAC* is defined in paragraph 3.5.

3.2 IPOPT

IPOPT is a code developed by Andreas Wächter, a chemical engineer, specifically designed to solve non-convex, nonlinear, continuous and smooth optimization problems also called Non Linear Programming (NLP) [4]. The code was initially designed in FORTRAN and then translated a couple of years later in C++ and it is an Open-source project.

3.3 Sequential Approximate Optimization

Sequential Approximate Optimization (SAO) has been selected for the purpose of this study because the number of variables in the simulation is considerably high and other methods would result computationally expensive.
Considering the k^{th} iteration with *l* the number of equality constraints, *m* the number of inequality constraints, *n* the number of design variables, $\bar{f}_0^k$ is the approximated function of f_0 and $\bar{g}_0^k$

$\bar{h}_0^k$ are the approximated functions of the equality and inequality constraints; the problem can be expressed as follows:

$$
\begin{gathered}
minimize\ \ \bar{f_0^k}(x) \\
\begin{cases}
\bar{g}_0^k(x) = 0,\ j = 1, ..., l \\
\bar{h}_0^k(x) \leq 0,\ j = 1, ..., m \\
\alpha_i^k \leq x_i \leq \beta_i^k\ i = 1, ..., n
\end{cases}
\end{gathered}
\tag{3.3}
$$

Where *x* is the vector containing the design variables, *k* is the number of design variables and α and β are the limits for the optimization of the considered design variables at the considered iteration.

3.4 Computational Structure

The process of optimization requires to continuously analyze the structure in order to update the results. The process is controlled by Matlab that performs the optimization but in order to compute the structure fundamental frequencies and the related displacement, the use of Abaqus is required.
Matlab communicates to Abaqus using Python scripts so the whole process is automatized and at every iterations the information is stored in *.txt* files in order to prevent data losses in case of unexpected crashes.

The computational process starts from a Matlab code that launches the Finite Element Analysis (FEA) analysis of the full scaled model using a pre-compiled python code that contains all the information needed by the finite element code to run. After convergence of the simulation of the full scale model, the initial design is updated and the pre-processing is completed.
When the pre-processing is completed the optimization loop can start and it is composed by three separate main sequences:

- Scaled model FEA analysis
- Sensitivity analysis
- IPOPT

The scaled model analysis is performed in the same way it is performed in the pre-processing, but differently from the full scale model analysis, it is performed every time the design is updated and therefore every optimization loop. After the simulation has converged the results of

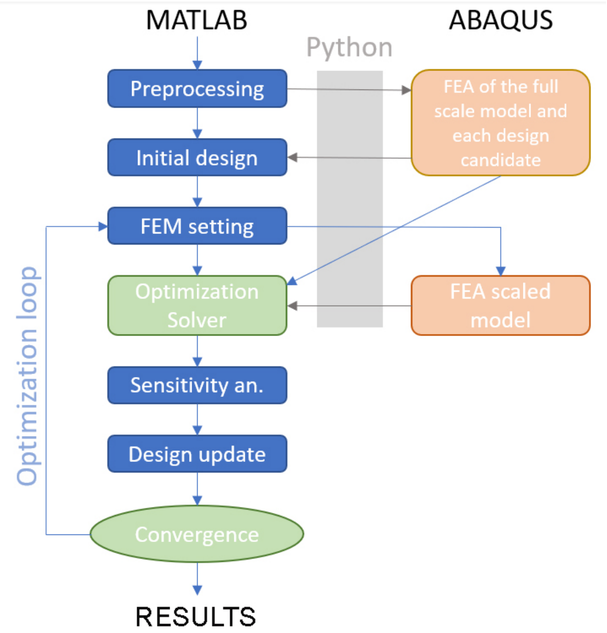


Figure 3.1: Computational Structure

the simulation is imported in the Matlab environment in the form of global variables in order to share them with the various functions of the code. The variables imported are essentially the eigenfrequences and eigenvectors of the structure and the stiffness matrices that are needed in order to perform the sensitivity analysis. Once the sensitivity analysis is completed the objective function, it's derivative, the inequality constraints and their derivatives are imported into IPOPT and the code performs the optimization that leads to a new material distribution.
The optimization loop is then iterated until the maximum number of iterations is reached.

3.5 Modal Assurance Criterion

The MAC is a statistical indicator sensible to the differences between mode shapes. In general it can be used in order to compare:

- Experimental eigen-vectors and analytical mode shapes.
- A defined mode shape with a modified one.

It is important in our case because it allows the code to directly compare the mode shapes of the full scaled wing-box and of the scaled wing-box; when comparing a set of mode shapes it returns an array of values between 0 and 1 where 1 means complete similarity and 0 means total difference but values over 0.9 already indicate a consistent correspondence and are acceptable for the purpose of the test [44]. A graphical example of the MAC calculation for 5 mode shapes is presented in figure 3.2.

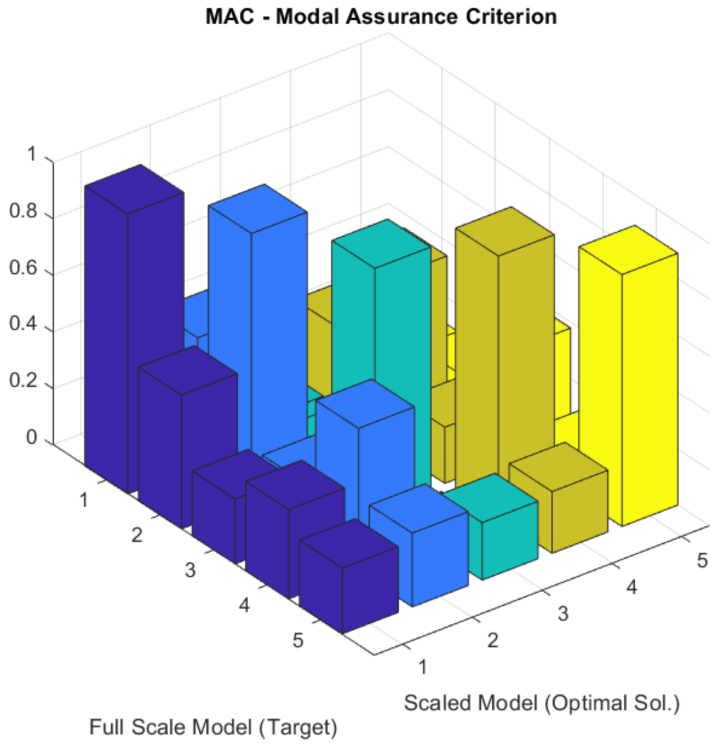


Figure 3.2: Modal Assurance Criterion

3.6 Sensitivity analysis

The sensitivity analysis plays a fundamental role in this thesis, the calculation of the sensitivity of the MAC numbers is required, but in order to perform this calculation it is required to calculate the sensitivity of the eigenvector. Initially the calculation of the sensitivity of the eigenvector was performed through Dailey's method that required the knowledge of the sensitivity of the eigenvalues but had the advantage that was able to calculate the derivatives of multiple eigenvalues. Unfortunately the implementation of this method did not succeed and a different method, called Adjoint method was used [45]. Both methods are going to be presented in the following pages after definition of the calculation of the eigenvalue derivatives.

3.6.1 Eigenvalue derivatives

The derivative of the eigenvalues is calculated considering the equation:

$$(K - \omega_s^2 M)\phi_\mathbf{s} = 0 \;\forall\; s = 1, ..., n_{dof} \tag{3.4}$$

Where *dof* is the number of eigenvalues considered. Differentiating 3.4 we obtain:

$$(\frac{\partial K(x)}{\partial x} - (\omega_s^2 \frac{\partial M(x)}{\partial x} + \frac{\partial \omega_s^2(x)}{\partial x} M))\phi_\mathbf{s} + (K - \omega_s^2 M)\frac{\partial \phi_\mathbf{s}}{\partial x} = 0 \tag{3.5}$$

Multiplying and reordering 3.5 by ϕ_s^T:

$$\phi_\mathbf{s}^\mathbf{T}(\frac{\partial K(x)}{\partial x} - (\omega_s^2 \frac{\partial M(x)}{\partial x} + \frac{\partial \omega_s^2(x)}{\partial x} M))\phi_\mathbf{s} = -\phi_\mathbf{s}^\mathbf{T}(K - \omega_s^2 M)\frac{\partial \phi_\mathbf{s}}{\partial x} \tag{3.6}$$

According to 3.4 the right part of 3.6 is zero and after some manipulation we obtain:

$$\frac{\partial \omega_s^2(x)}{\partial x} = \frac{\phi_s^T(\partial \frac{K(x)}{\partial x} - \omega_s^2 \frac{\partial M(x)}{\partial x})\phi_s}{\phi_s^T M \phi_s} \tag{3.7}$$

Assuming that the eigenvector is mass-normalized the denominator of the right part of the equation is equal to 1:

$$\frac{\partial \omega_s^2(x)}{\partial x} = \phi_s^T(\partial \frac{K(x)}{\partial x} - \omega_s^2 \frac{\partial M(x)}{\partial x})\phi_s \tag{3.8}$$

And the derivative of the eigenfrequency can be defined:

$$\frac{\partial \omega_s(x)}{\partial x} = \frac{1}{2\omega_s}\frac{\partial \omega_s^2(x)}{\partial x} \tag{3.9}$$

3.6.2 Dailey's method

Several methods have been proposed in order to calculate the eigenvector derivatives, Dailey's method was selected because it can consider also cases where the eigenvalues associated to the eigenvectors are multiple. This method is derived from Nelson's method in order to include the case of repeated eigenvalues and from this method the Friswell algorithm has been developed [46]. The algorithm used can be summarized in 10 steps:

1. D = $X^T(K' - \lambda M')X$
2. Solve the eigenvalue problem $D\Gamma = \Gamma\Lambda'$ where Λ is the diagonal matrix of the eigenvalue derivatives calculated by 3.9 and Γ should be normalized.

3. The columns of $Z = X\Gamma$ become the new eigenvectors.

4. Compute $G = K - \lambda M$ and $f = (\lambda M' - K')Z + MZ\Lambda'$

5. Find the position of the largest element in Z and substitute in the corresponding columns and rows of G and rows of f the number 0. In the diagonal position of G affected by the substitution replace with 1. The new matrices are called $\bar{G}$ and $\bar{f}$

6. Solve $\bar{G}$V= $\bar{f}$

7. $Q = -V^T MZ - Z^T MV - Z^T M'Z$

8. $R = Z^T(K' - \lambda M')V - Z^T(M'Z + MV)\Lambda' + 0.5Z^T(K'' - \lambda M'')Z$

9. Build and mxm matrix C in the following way:

$$c_{ij} = \begin{cases} r_{ij}/(\lambda'_j - \lambda'_i) & if \quad \lambda'_j \neq \lambda'_i \\ 0.5q_{ij} & otherwise \end{cases}$$

10. Calculate $Z' = V + ZC$ where the columns of Z' are the eigenvector derivatives

3.6.2.A MAC derivatives in the Dailey's Method

The MAC derivatives are finally calculated using the eigenvectors of both structures and their derivatives [45]:

$$\frac{\partial MAC}{\partial \rho} = [\frac{2(\phi_A^T \phi_X)^2}{(\phi_A^T \phi_A)(\phi_Q^T \phi_Q)^2}\phi_Q^T]\frac{\partial \phi_Q}{\partial \rho} \tag{3.10}$$

where ϕ_A is the eigenvector related to the full scale model and ϕ_X is the eigenvector related to the scaled model.

3.6.3 Adjoint method

The adjoint method developed in this work is presented by Tsai ad Cheng [45], it requires five steps in order to be performed, the eigenvectors of the two structures and the global matrices of the scaled model are required as an input. It is important to remark that the method cannot distinguish the eigenvectors in the case of repeated eigenvalues. The solution adopted by Tsai is to maximize the natural frequencies, which provides an easy and straightforward way to prevent mode switching [45], that is not applicable in this case study and therefore extra attention needs to be used to monitor mode switching.

1. $a = ((\frac{(\psi_0^T \psi_j)}{(\psi_0^T \psi_0)(\psi_j^T \psi_j)})\psi_0^T - (\frac{(\psi_0^T \psi_j)^2}{(\psi_0^T \psi_0)(\psi_j^T \psi_j)^2})\psi_j^T)\psi_0^T$

2. $c = ((\frac{(\psi_0^T \psi_j)}{(\psi_0^T \psi_0)(\psi_j^T \psi_j)})\psi_0^T - (\frac{(\psi_0^T \psi_j)^2}{(\psi_0^T \psi_0)(\psi_j^T \psi_j)^2})\psi_j^T) - 2a\psi_0^T M$

3. $\alpha_p = \frac{K - \lambda M}{c}$

4. $b = -\alpha_p M \psi_0^T$

5. $\alpha_j = b\psi_0^T + \alpha_p$

3.6.3.A MAC derivatives in the Adjoint Method

The MAC derivatives in the adjoint method are calculated using the global mass and stiffness matrices of the scaled model, they are updated every iteration and used as a derivative of the inequality constraint inside the IPOPT optimize. Since the global matrices are not normally given as an output by Abaqus, a new instance of Abaqus has to opened ever iteration adding computational time. The steps used in the calculation are the following, α_j is then used by the equation 3.11 to calculate the derivatives of the Modal Assurance Criterion:

$$\frac{\partial MAC}{\partial \rho} = \alpha_j \frac{\partial K}{\partial \rho}\psi_j + (a\psi_j - \lambda\alpha_j)\frac{\partial K}{\partial \rho}\psi_j \tag{3.11}$$

3.7 Wing-box

The case in exam uses a structure, called wing box, that is an idealization of a real wing-box present inside the skin of the wing in a normal aircraft. This element is designed to provide the necessary strength in the lightest way possible and usually it is composed by numerous ribs and spars.

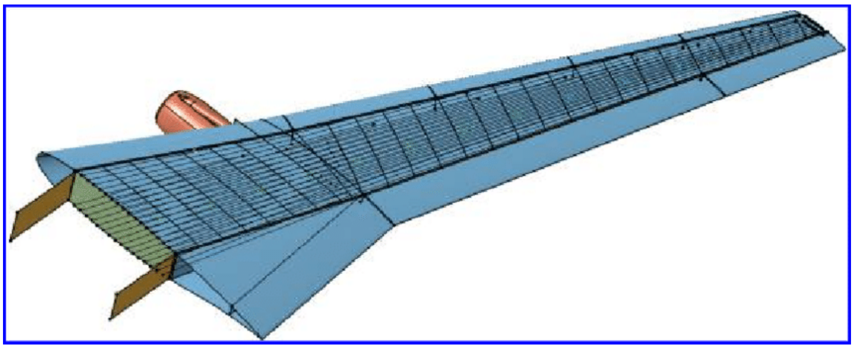


Figure 3.3: Wing-box model [47]

The wing-box used, being a simplification, is a sandwich structure. This type of structures are composed by two thin and strong skins divided by a thick layer of lightweight core that can be a foam or an honeycomb. The core main function is to separate the two skins and carry the load from one skin to the other and, while slightly increasing the weight, it dramatically increases the inertia of the whole structure improving bending and buckling loads [48].

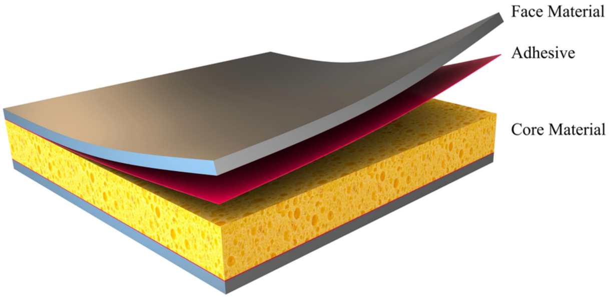
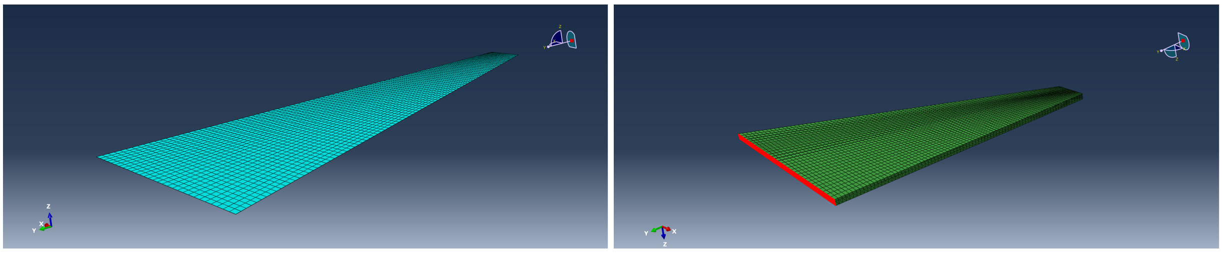


Figure 3.4: Sandwich model [49]

The main advantage related to the adoption of a continuous media between the two skins is the ability to discretize the core in small parallelepipeds and to vary the material properties of every element in order to optimize the structure and obtain the final design.

3.8 Abaqus Model



(a) Skin mesh

(b) Core mesh with boundary conditions

Figure 3.5: Abaqus generated mesh for the wing-box

The 3D model for the finite element analysis is presented in figure 3.5, the dimensions of the box are reported in table 3.1. The scaling factor between the two models is 10 and therefore the natural frequencies and the modal displacement will need to be scaled by a factor of 10 in order to be directly compared.

Instance	Full Scale Model [mm]	Scaled Model [mm]
Base Chord	3200	320
Tip Chord	500	50
Half Wingspan	4600	460
Thickness	100	10

Table 3.1: Model dimensions comparison

The mesh of the skin is composed by shell elements, and the mesh of the internal core is globally composed by 19200 hexhaedral 3D elements divided in 8 layers of 2400 cells each. Only the internal core is included in the optimization process while the skin material distribution does not change during the optimization.

The great computational time required to perform the optimization however requires to simplify the model and reduce the number of cells used, the patch approach is therefore used.

The patch used in the optimization is applied at a layer level, every patch contains 12 cells and the material properties of the sub-cells are the same if they are shared by the same patch.

4

Results

Contents

4.1 Multi Material Optimization

MMTO is a computationally complex and heavy optimization process. As introduced in chapter 3 the framework requires to perform a finite element analysis every iteration that is performed using Abaqus. This simulation requires, with a modern CPU equipped with six cores running at 4.2GHz, around 400 seconds.

Apart from this, the Matlab code requires 450 seconds in total to perform the various tasks required, including the calculation of the sensitivity of the eigenvectors but excluding the IPOPT optimizer itself.

The aforementioned optimizer time limit is set to 500 seconds and allows it to perform around 130 sub-iterations.

Instance	Time [s]	Total time [s]
Abaqus	400	
Matlab	450	
IPOPT	500	1350

Table 4.1: Computational time

This premise, even if not required, is important to specify why only a limited number of simulations is presented in this thesis. A more powerful cluster would allow for a faster implementation of different solutions.

The biggest difficulty behind the implementation of new alternatives for the code is that the total CPU time to complete a global iteration takes 1350 seconds or around 22 and a half minutes. Considering that in order to understand if the results are converging to a feasible result the simulation has to perform preferably 200 iterations this means that a simulation ends after more than three days without considering predictable crashing that fortunately can be recovered by the code.

For the reasons presented, five different case-studies of the wing-box are presented, where the material properties of the multi-material core are changed together with the penalty factor and the filter application.

4.1.1 Case 1

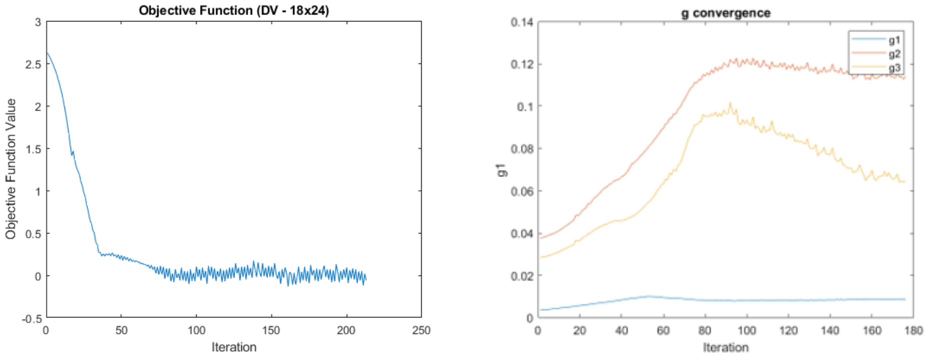
Penalization factor = 2.4			
Material	E [GPa]	$\rho[kg/m^3]$	ν
Core (Full Scale)	0.03	1000	0.32
Skin (Full Scale)	70	2700	0.33
Core 1 (Scaled) 🟨	4	1180	0.3
Core 2 (Scaled) 🟦	1	1150	0.3
Core 3 (Scaled) 🟩	$1e^{-8}$	10	0.3
Skin (Scaled)	70	2700	0.33

Table 4.2: Material properties of case 1

The first case to be presented for the optimization considers two different materials for the core, the two material selected present a Young modulus of 4 and 1 respectively while the third material is the void that can't be represented with a value of 0 in order to avoid numerical errors. The feasibility of the part is respected even with good FDM printers since it is possible to find in commerce nylon based polymers that respect this data.

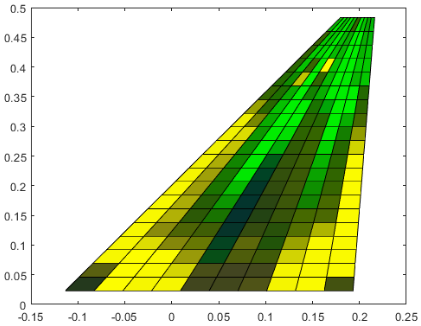
The results show a good converge of the objective function while the constraint function related to the MAC of the first three modes is maintained low with the first mode in particular showing a very low value of 0.009 and therefore a great similarity.

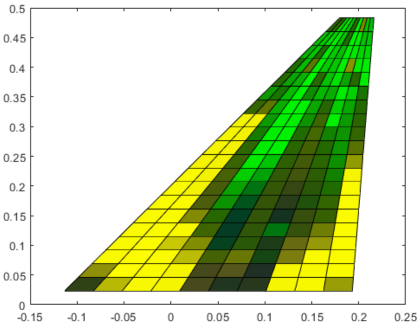
The eight layers, represented on the next page, show a disordered distribution of material with some formation of checkerboard patterns. It was decided to perform other simulations with different materials and penalization factors in order to investigate the impact of those parameters on the final results.



(a) Objective function convergence **(b)** Constraint function convergence

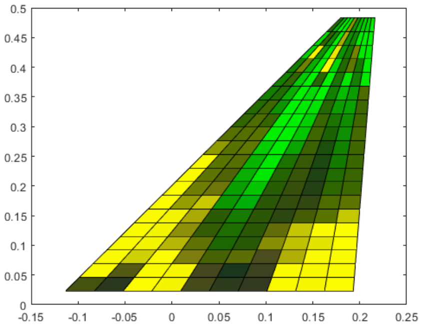
Figure 4.1: Case 1

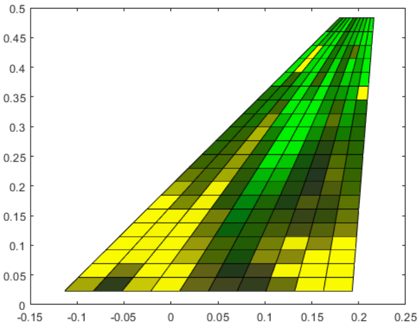




(a) Layer 1

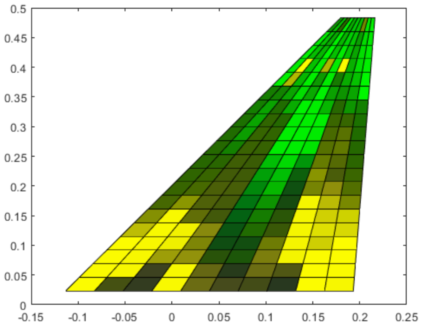
(b) Layer 2

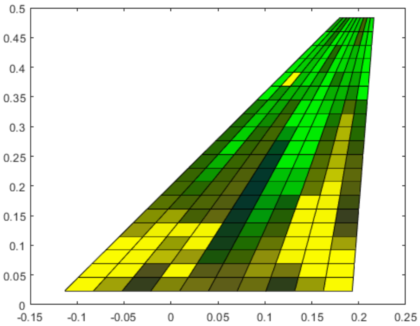




(c) Layer 3

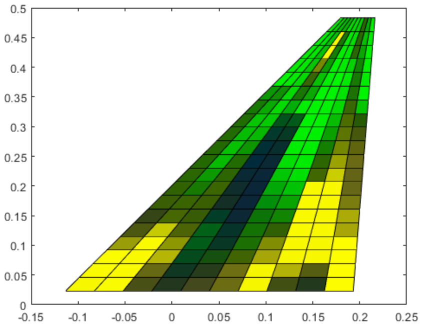
(d) Layer 4

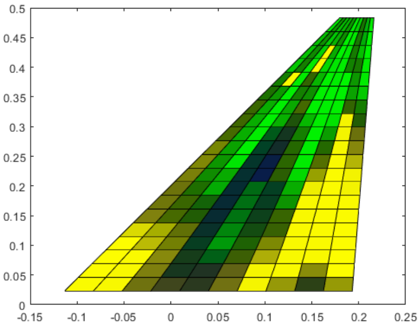




(e) Layer 5

(f) Layer 6





(g) Layer 7

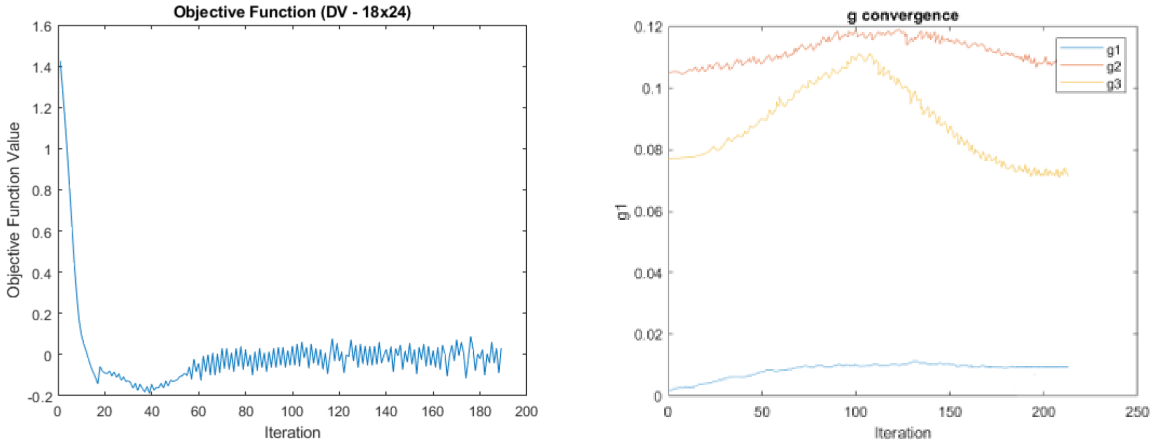
(h) Layer 8

4.1.2 Case 2 (increased penalization factor)

Penalization factor = 3				
Material		E [GPa]	$\rho [kg/m^3]$	ν
Core (Full Scale)		0.03	1000	0.32
Skin (Full Scale)		70	2700	0.33
Core 1 (Scaled)	🟨	4	1180	0.3
Core 2 (Scaled)	🟦	1	1150	0.3
Core 3 (Scaled)	🟩	$1e^{-8}$	10	0.3
Skin (Scaled)		70	2700	0.33

Table 4.3: Material properties of case 2

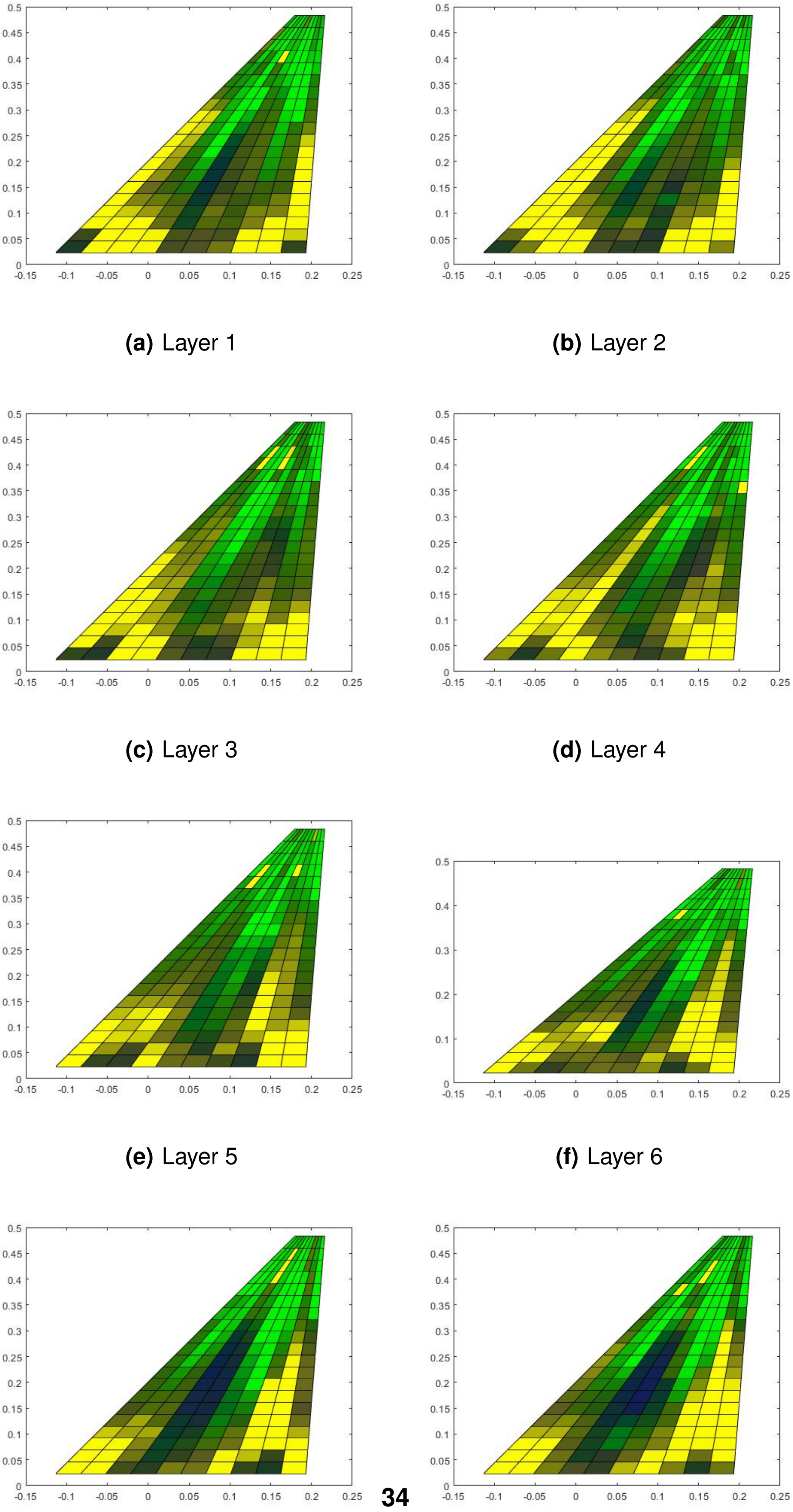
This case is very similar to case number one. The only difference with it is the increase of the penalization factor that, as presented previously, tends to limit the formation of intermediate densities. From the results, this tendency is not very ascertainable because the variation of the penalization factor is quite small but it is interesting to notice the formation of a blue zone in the last two plies that represents material number 2.



(i) Objective function convergence

(j) Constraint function convergence

Figure 4.2: Case 2



(a) Layer 1

(b) Layer 2

(c) Layer 3

(d) Layer 4

(e) Layer 5

(f) Layer 6

(g) Layer 7

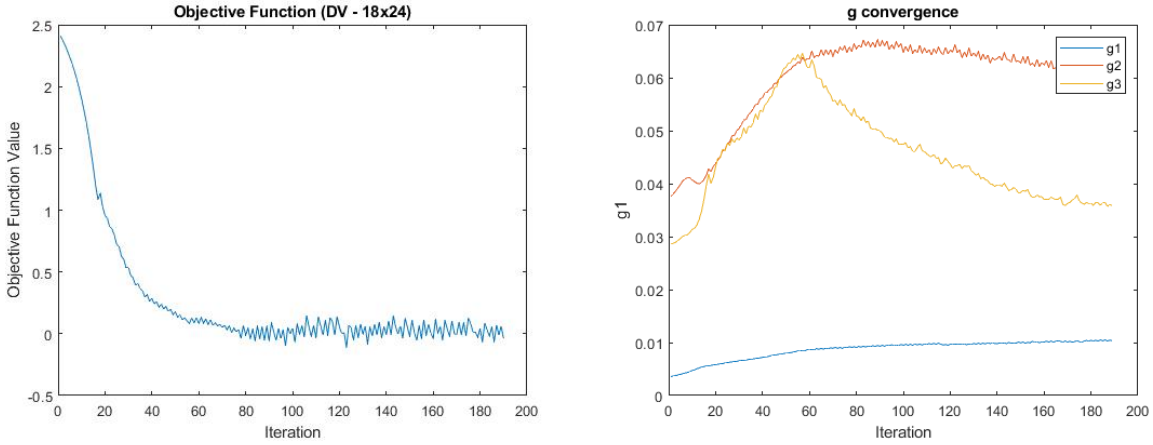
(h) Layer 8

4.1.3 Case 3 (Material Variation)

Penalization factor = 3			
Material	E [GPa]	$\rho[kg/m^3]$	ν
Core (Full Scale)	0.03	1000	0.32
Skin (Full Scale)	70	2700	0.33
Core 1 (Scaled) 🟨	4	1180	0.3
Core 2 (Scaled) 🟦	3	1150	0.3
Core 3 (Scaled) 🟩	$1e^{-8}$	10	0.3
Skin (Scaled)	70	2700	0.33

Table 4.4: Material properties of case 3

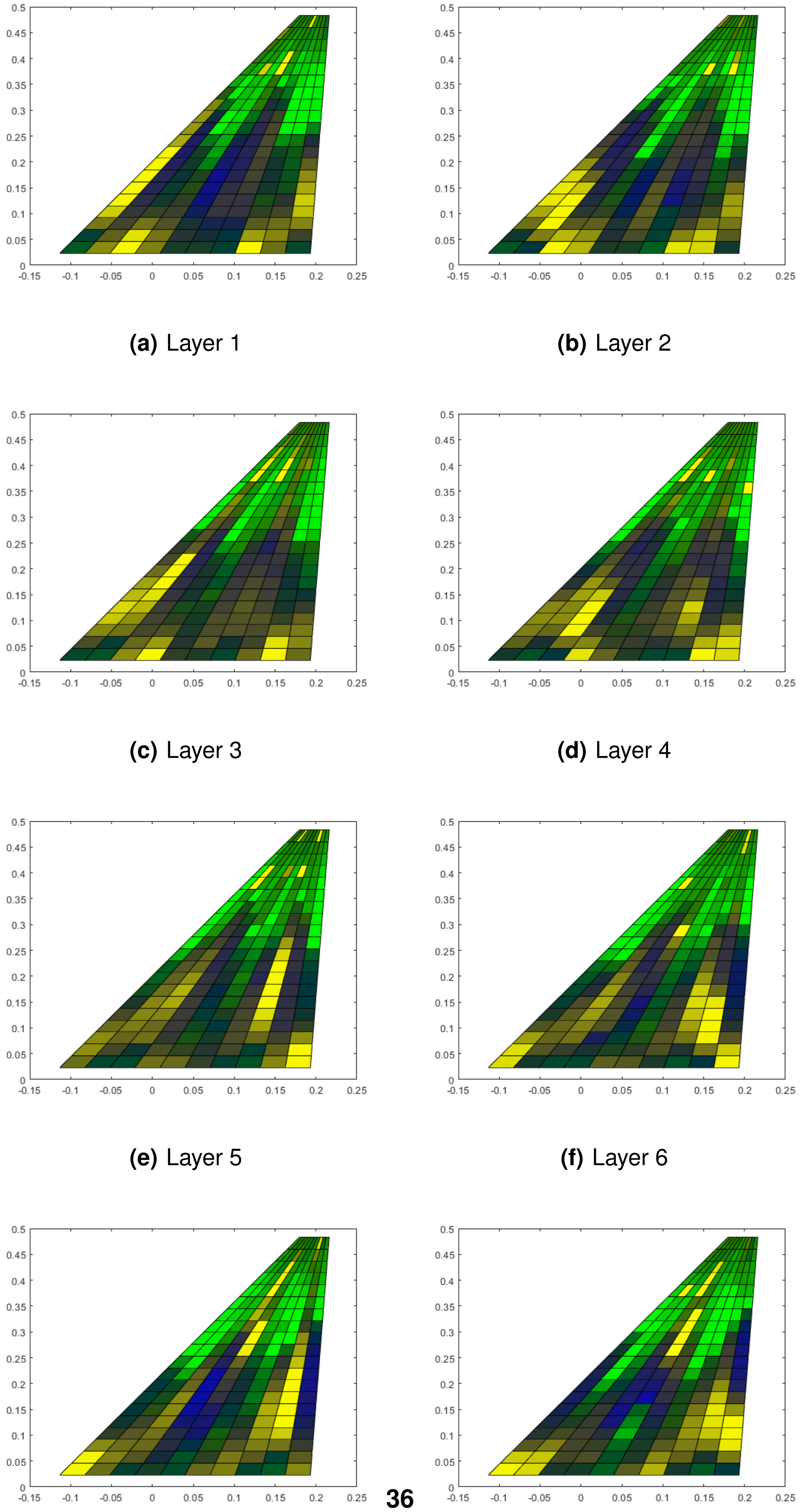
Case 3 and 4 differently from the first two cases maintain the penalty factor equal to 3 but variate the material properties. In this case the material properties are chosen to be more similar to each other with material number two that presents a Young modulus of 3 with a ratio between the two material of 1.33 compared to the previous ratio of 4. The results show that material number two, that in the first two simulation was nearly absent, now is present but it is also possible to notice a great number of patches with intermediate densities.
It is also important to notice that in figure 4.3 the curves related to mode number 2 and 3 overlap each other displaying a possible problem of mode swapping.



(i) Objective function convergence

(j) Constraint function convergence

Figure 4.3: Case 3



(a) Layer 1

(b) Layer 2

(c) Layer 3

(d) Layer 4

(e) Layer 5

(f) Layer 6

(g) Layer 7

(h) Layer 8

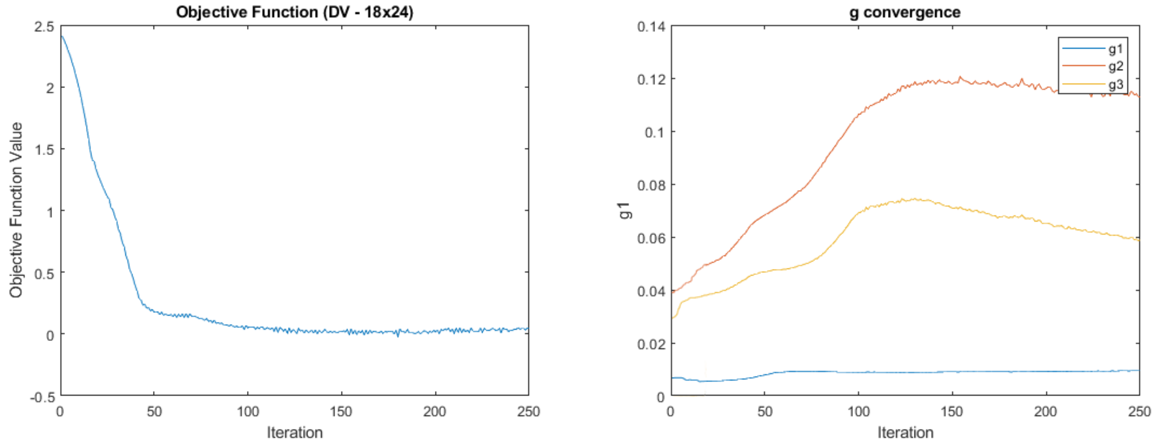
4.1.4 Case 4 (Material Variation)

Penalization factor = 3			
Material	E [GPa]	$\rho [kg/m^3]$	ν
Core (Full Scale)	0.03	1000	0.32
Skin (Full Scale)	70	2700	0.33
Core 1 (Scaled) 🟨	8	1180	0.3
Core 2 (Scaled) 🟦	1	1150	0.3
Core 3 (Scaled) 🟩	$1e^{-8}$	10	0.3
Skin (Scaled)	70	2700	0.33

Table 4.5: Material properties of case 4

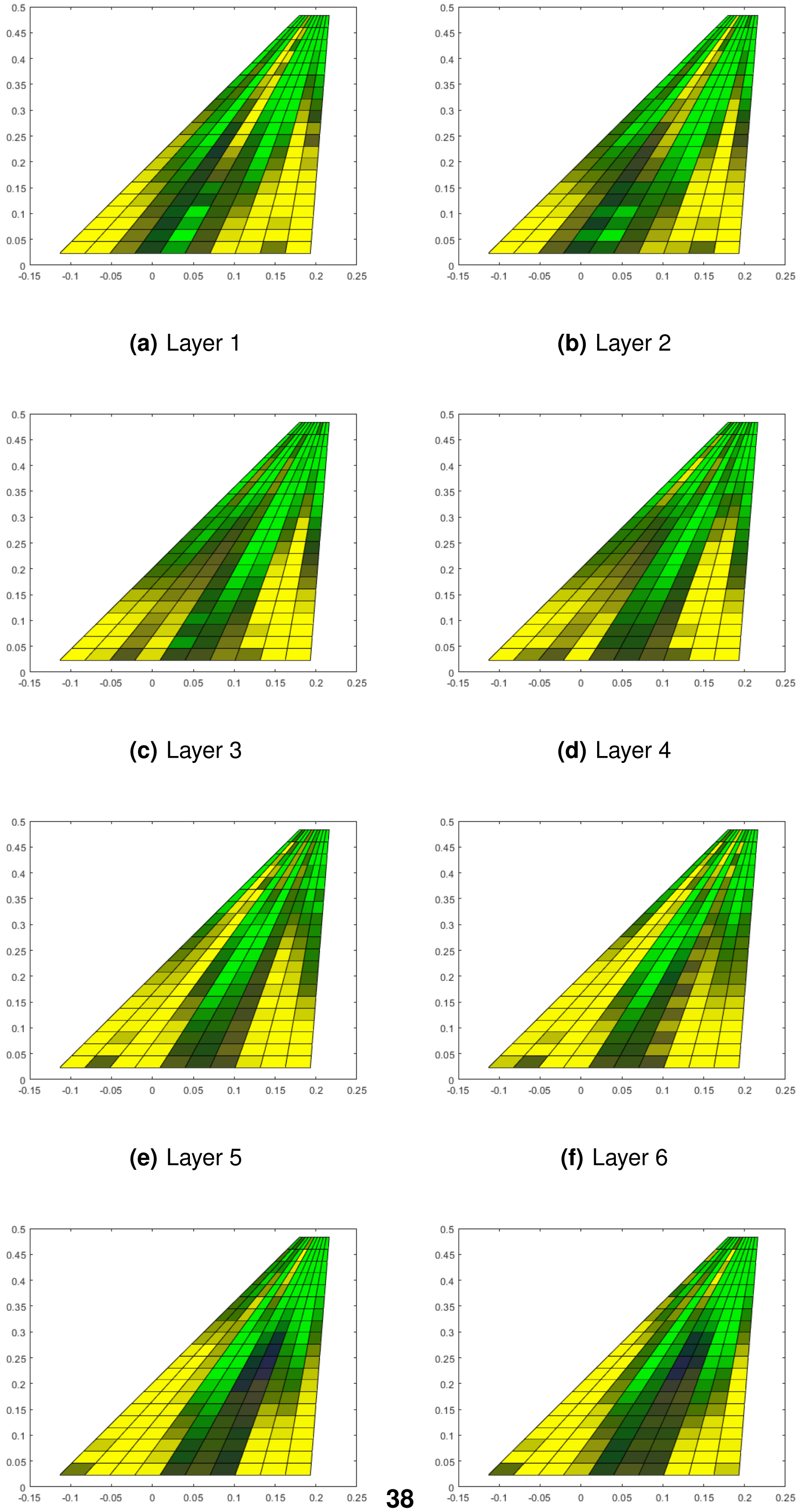
As introduced before, simulation number four uses a different material ratio between core 1 and 2. The ratio is equal to 8 and this provides some interesting results that should be considered for future developments.

In the pictures presented on the next page it is possible to see that material number two is avoided by the code and only material number one is used. The distribution is good in terms of intermediate densities but begins to develop traces of checkerboard patterns especially in the sixth layer.



(i) Objective function convergence **(j)** Constraint function convergence

Figure 4.4: Case 4



(a) Layer 1

(b) Layer 2

(c) Layer 3

(d) Layer 4

(e) Layer 5

(f) Layer 6

(g) Layer 7

(h) Layer 8

4.1.5 Case 5 (Density Filter Application)

Penalization factor = 3			
Material	E [GPa]	$\rho [kg/m^3]$	ν
Core (Full Scale)	0.03	1000	0.32
Skin (Full Scale)	70	2700	0.33
Core 1 (Scaled) 🟨	4	1180	0.3
Core 2 (Scaled) 🟦	1	1150	0.3
Core 3 (Scaled) 🟩	$1e^{-8}$	10	0.3
Skin (Scaled)	70	2700	0.33

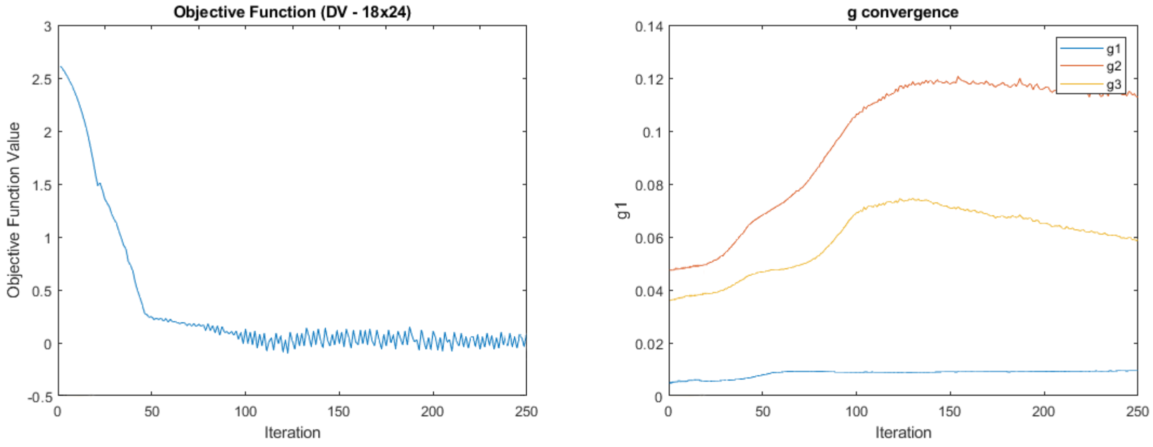
Table 4.6: Material properties of case 5

The last case presented uses the same materials of case one and two but with the application of a density filter. The filter used can be expressed by equation 4.1:

$$\vec{f_e} = \frac{\sum H_{ei} f_i}{\sum H_{ei}} \tag{4.1}$$

Where $H_{ei} = max(0, r_m - \Delta(e, i))$ and r_m=1.7. The filter is applied to the material densities and its function is to average the densities of the contiguous cells in a layer.

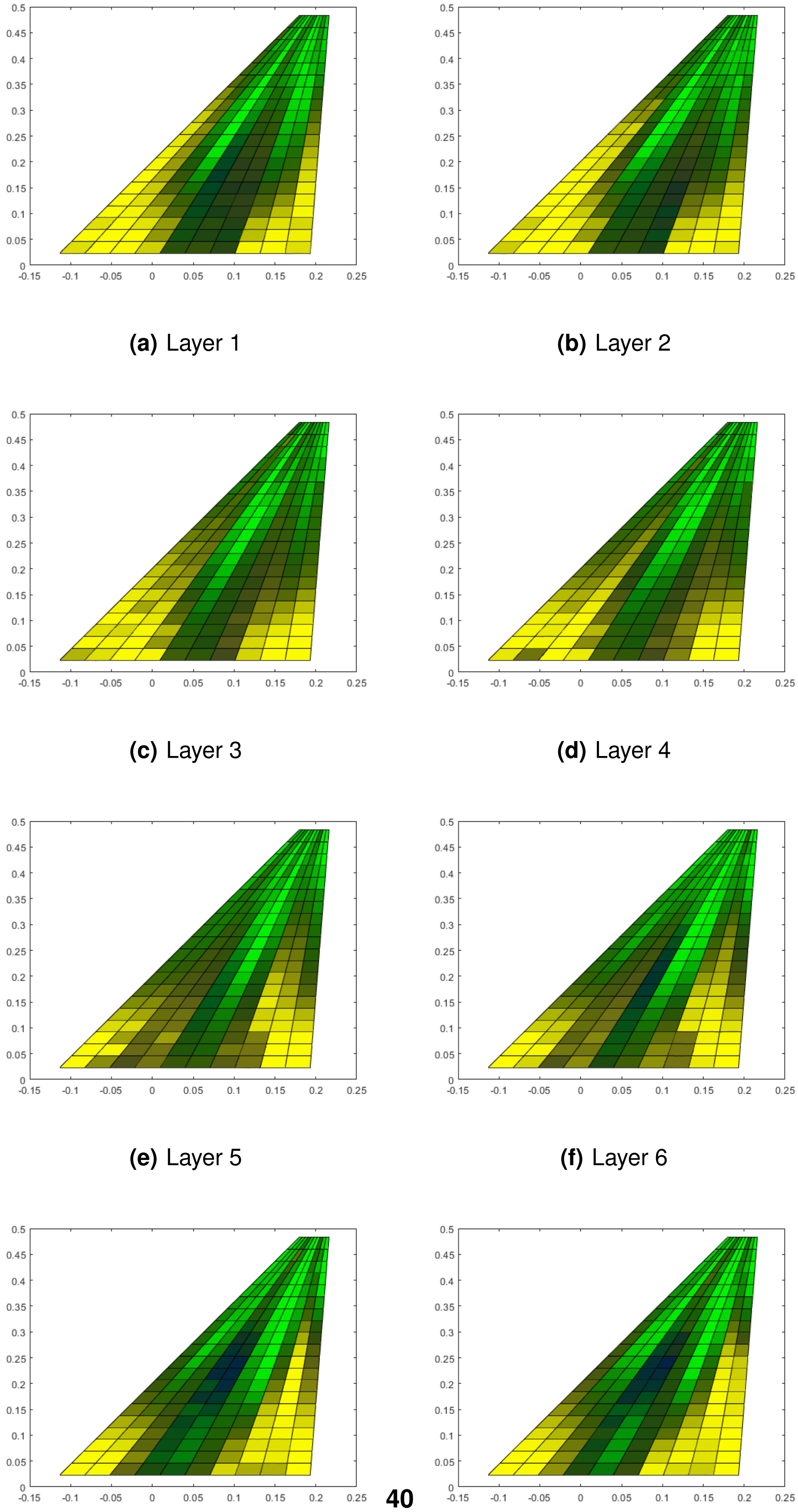
The results obtained are going to be analyzed more in detail after the presentation of the layers in the following page.



(i) Objective function convergence

(j) Constraint function convergence

Figure 4.5: Case 5



(a) Layer 1

(b) Layer 2

(c) Layer 3

(d) Layer 4

(e) Layer 5

(f) Layer 6

(g) Layer 7

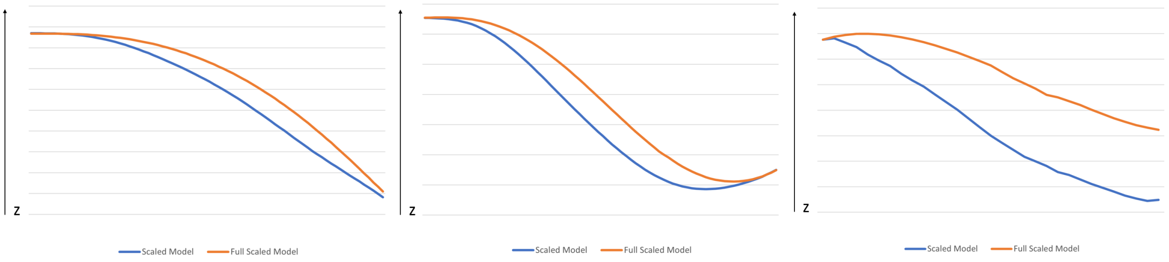
(h) Layer 8

The first and the second mode are bending modes and their natural frequencies have to be scaled by the geometrical scaling factor of 10, the results are presented in table 4.7 and show a very good comparability with a maximum percentual difference of 0.86 for the first three modes. This result was expectable considering the plot of the objective function that converged to a really close value of zero.

Mode	Scaled	Full scale	Full scale *10	% difference
1	34.89	3.52	35.19	0.86
2	111.08	11.16	111.55	0.43
3	137.67	13.87	138.66	0.72

Table 4.7: Natural frequency comparison

The mode comparison as described in the third chapter has been done with the calculation of the MAC number and in order to visualize the results, a scaled plot of the displacements in the z direction is presented in figure 4.6.

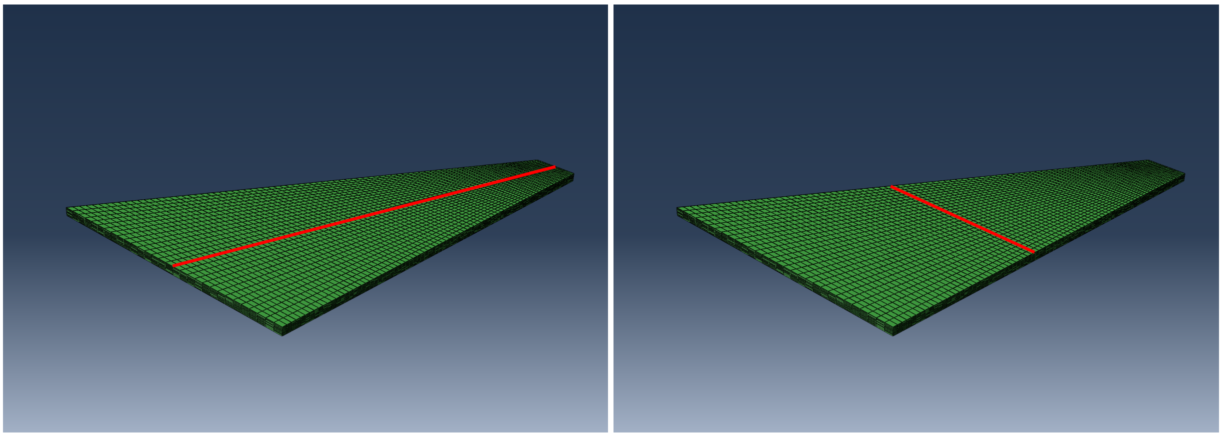


(i) First mode (bending) **(j)** Second mode (bending) **(k)** Third mode (torsion)

Figure 4.6: Visual mode comparison

The nodes designated for the plot of the displacements for the first two modes (bending modes) are represented in figure 4.7(a) and are selected along the longitudinal mid-line while the nodes designated for the displacements of the third mode (torsional mode) are presented in figure 4.7(b).

The two bending modes are well replicated in the scaled model and this is also shown by a the MAC number related to the two modes in figure 4.7(b) of 0.982 for the first mode and 0.920 for the second mode. The cross member that compare the first and the second mode is also high and this is identifiable in the similarity that the two modes have. The third mode is not represented in the proper way and this is underlined also by the MAC value that is lower than 0.9, a value that should be considered minimum for the mode shape similarity [44].



(a) Node selection (bending)

(b) Node selection (torsion)

Figure 4.7: Node displayed in plot 4.6

5

Conclusion

Topology optimization is nowadays included in many commercial codes because the industry understood its capability in terms of performance and thanks to the additive manufacturing increased popularity the designs produced are feasible and progressively more economic.
The largest part of those designs are focused on weight saving and dynamic scaling is still not considered but numerous applications could benefit of this technology. The main goal of this thesis was to develop an algorithm that allowed to include in the code the calculation of the inequality constraints sensitivity and therefore guide the optimization to obtain a dinamically scaled model.
Using the inequality constraints sensitivity in the form of MAC derivatives, the code performs a multi-material topology optimization that has the target of producing a dynamically scaled model with matching eigenvalues and eigenfrequencies. In order to perform this calculation the Dailey's method and the Adjoint method have been implemented inside the Matlab environment. Adjoint method was then selected in order to perform the simulations because it's implementation displayed better results.
Adjoint method directly calculates the MAC derivatives using the global K and M matrices of the scaled model, it's implementation is simple and the calculation is quite fast but the main issue related to this method is that it doesn't consider the case of repeated eigenvalues and therefore this aspect has to be taken in consideration.
The results obtained, displayed in chapter 4, where multiple designs were analyzed, show that the code was able to produce designs with eigenfrequencies comparable to the one of the full scaled model with errors under 1%. The first and the second eigenmodes have been represented quite carefully by every simulation but the third mode, a torsional mode, still displayed need of an improvement.
The results could be improved increasing the number of patches and therefore the computational time but this was not feasible with the setup used for the thesis.
The drawback that the simulation displayed is that the influence of multiple materials have not made a huge impact in the results since most of the final designs preferred the first material, the stiffest one, than the second. This behaviour has to be investigated with the selection of different materials.

For the future developments of the work some aspects should be underlined:

- Ulterior simulations with an higher number of patches should be performed in order to obtain a better fit for the third vibrational mode.
- The application of different filtering techniques in order to obtain a smoother solution and allow the 3D printed manufacturing of a real model.

- The introduction of lattice cells in the optimization in order to allow intermediate designs in the 3D printing.
- To physically produce the final design and to test its dynamical properties in a wind tunnel in order to validate the model. This implementation is feasible for two main reasons, the size of the scaled model could potentially fit in a large number of wind tunnels and the materials used in the implementations are chosen with material properties that are close to the ones of polymers available for FDM printers.

Bibliography

[1] M. P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering*, vol. 71, no. 2, pp. 197–224, 1988. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0045782588900862

[2] K. C. Nguyen, P. Tran, and H. X. Nguyen, "Multi-material topology optimization for additive manufacturing using polytree-based adaptive polygonal finite elements," *Automation in Construction*, vol. 99, pp. 79–90, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580518309518

[3] M. Zhou and G. Rozvany, "The coc algorithm, part ii: Topological, geometrical and generalized shape optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 89, no. 1, pp. 309–336, 1991, second World Congress on Computational Mechanics. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0045782591900469

[4] A. Wächter, "An interior point algorithm for large-scale nonlinear optimization with applications in process engineering," Ph.D. dissertation, 2002.

[5] T. D. Tsai and C. C. Cheng, "Structural design for desired eigenfrequencies and mode shapes using topology optimization," *Struct. Multidiscip. Optim.*, vol. 47, no. 5, p. 673–686, may 2013. [Online]. Available: https://doi.org/10.1007/s00158-012-0840-2

[6] R. Nelson, "Simplified calculation of eigenvector derivatives," *AIAA Journal*, vol. 14, pp. 1201–1205, 1976.

[7] R. L. Dailey, "Eigenvector derivatives with repeated eigenvalues," *AIAA Journal*, vol. 27, no. 4, pp. 486–491, 1989. [Online]. Available: https://doi.org/10.2514/3.10137

[8] M. I. Friswell, “The Derivatives of Repeated Eigenvalues and Their Associated Eigenvectors,” *Journal of Vibration and Acoustics*, vol. 118, no. 3, pp. 390–397, 07 1996. [Online]. Available: https://doi.org/10.1115/1.2888195

[9] S. A. Tofail, E. P. Koumoulos, A. Bandyopadhyay, S. Bose, L. O’Donoghue, and C. Charitidis, “Additive manufacturing: scientific and technological challenges, market uptake and opportunities,” *Materials Today*, vol. 21, no. 1, pp. 22–37, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1369702117301773

[10] B. Nagesha, V. Dhinakaran, M. Varsha Shree, K. Manoj Kumar, D. Chalawadi, and T. Sathish, “Review on characterization and impacts of the lattice structure in additive manufacturing,” *Materials Today: Proceedings*, vol. 21, pp. 916–919, 2020, international Conference on Recent Trends in Nanomaterials for Energy, Environmental and Engineering Applications. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214785319331451

[11] O. Sigmund and K. Maute, “Topology optimization approaches a comparative review,” *Structural and Multidisciplinary Optimization*, vol. 48, 12 2013.

[12] P. Pallarès, “Comparative study of the goodness of topology optimization software on additive manufactured parts,” Ph.D. dissertation, Universitat Politècnica de Catalunya, Campus Diagonal Sud, Edifici H. Av. Diagonal, 647 08028 Barcelona, 7 2020.

[13] M. Stolpe and K. Svanberg, “An alternative interpolation scheme for minimum compliance topology optimization,” *Structural and Multidisciplinary Optimization*, vol. 22, no. 2, p. 116–124, 2001.

[14] Y. Xie and G. Steven, “A simple evolutionary procedure for structural optimization,” *Computers e Structures*, vol. 49, no. 5, pp. 885–896, 1993. [Online]. Available: https://www.sciencedirect.com/science/article/pii/004579499390035C

[15] O. Querin, G. Steven, and Y. Xie, “Evolutionary structural optimisation using an additive algorithm,” *Finite Elements in Analysis and Design*, vol. 34, no. 3, pp. 291–308, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168874X9900044X

[16] X. Yang, Y. Xie, G. Steven, and O. Querin, “Bidirectional evolutionary method for stiffness optimization,” *AIAA journal*, vol. 37, no. 11, pp. 1483–1488, 1999.

[17] E. Biyikli and A. To, “Proportional topology optimization: A new non-gradient method for solving stress constrained and minimum compliance problems and its implementation in matlab,” *PLOS ONE*, vol. 10, 11 2014.

[18] W. Cheng, H. Wang, M. Zhang, and R. Du, “Improved proportional topology optimization algorithm for minimum volume problem with stress constraints,” *Engineering Computations*, vol. ahead-of-print, 06 2020.

[19] C. S. Jog and R. B. Haber, “Stability of finite element models for distributed-parameter optimization and topology design,” *Computer Methods in Applied Mechanics and Engineering*, vol. 130, no. 3, pp. 203–226, 1996. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0045782595009280

[20] A. Diaz and O. Sigmund, “Checkerboard patterns in layout optimization,” *Structural Optimization*, vol. 10, pp. 40–45, 08 1995.

[21] B. Bourdin, “Filters in topology optimization,” *International Journal for Numerical Methods in Engineering*, vol. 50, pp. 2143 – 2158, 03 2001.

[22] S.-B. Hu, L.-P. Chen, Y. Zhang, J. Yang, and S.-T. Wang, “A crossing sensitivity filter for structural topology optimization with chamfering, rounding, and checkerboard-free patterns,” *Structural and Multidisciplinary Optimization*, vol. 37, pp. 529–540, 02 2009.

[23] B. Bourdin, “Filters in topology optimization,” *International Journal for Numerical Methods in Engineering*, vol. 50, pp. 2143 – 2158, 03 2001.

[24] J. Stegmann and M. Stolpe, “Discrete material optimization of laminated composites - simp vs. global optimization,” 01 2006.

[25] C. Hull, “Apparatus for production of three-dimensional objects by stereolithography,” Mar 1986.

[26] G. Suresh, M. Reddy, G. Narendra, G. Santosh Kumar, and S. Balasubramanyam, “Summarization of 3d-printing technology in processing & development of medical implants,” *JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES*, vol. 14, pp. 176–191, 02 2019.

[27] L. Carolo and B. Obudho, “Tough resin amp; abs-like resin: Best brands of 2021,” Nov 2021. [Online]. Available: https://all3dp.com/2/best-tough-resin-abs-like-resin/

[28] T. Maruyama, H. Hirata, T. Furukawa, and S. Maruo, “Multi-material microstereolithography using a palette with multicolor photocurable resins,” *Optical Materials Express*, vol. 10, p. 2522, 10 2020.

[29] S. Crump, “Apparatus and method for creating three-dimensional objects,” Stratasys Inc., Jun 1992, patent Number: US5121329A.

[30] Z. Li, A. S. Rathore, C. Song, S. Wei, Y. Wang, and W. Xu, “Printracker: Fingerprinting 3d printers using commodity scanners,” in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1306–1323. [Online]. Available: https://doi.org/10.1145/3243734.3243735

[31] F. Moretti, “Delta wasp 3mt.” [Online]. Available: https://www.3dwasp.com/stampante-3d-cemento-delta-wasp-3mt-concrete/

[32] D. Jamie, “The manufacturers of 3d printed houses,” Mar 2020. [Online]. Available: https://www.3dnatives.com/en/3d-printed-house-companies-120220184/#!

[33] C. Höller, T. Hinterbuchner, P. Schwemberger, P. Zopf, R. Pichler, and F. Haas, “Direct machining of selective laser melted components with optimized support structures,” *Procedia CIRP*, vol. 81, pp. 375–380, 2019, 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827119303701

[34] L. Jiao, Z. Chua, S. Moon, J. Song, G. Bi, and H. Zheng, “Femtosecond laser produced hydrophobic hierarchical structures on additive manufacturing parts,” *Nanomaterials*, vol. 8, p. 601, 08 2018.

[35] C. Panwisawas, B. Perumal, R. M. Ward, N. Turner, R. P. Turner, J. W. Brooks, and H. C. Basoalto, “Keyhole formation and thermal fluid flow-induced porosity during laser fusion welding in titanium alloys: Experimental and modelling,” *Acta Materialia*, vol. 126, pp. 251–263, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359645416310102

[36] L. Hao, D. Raymont, C. Yan, A. Hussein, and P. Young, “Design and additive manufacturing of cellular lattice structures,” in *The International Conference on Advanced Research in Virtual and Rapid Prototyping (VRAP). Taylor & Francis Group, Leiria*, 2011, pp. 249–254.

[37] E. Alabort, D. Barba, and R. C. Reed, “Design of metallic bone by additive manufacturing,” *Scripta Materialia*, vol. 164, pp. 110–114, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359646219300399

[38] T. Maconachie, M. Leary, B. Lozanovski, X. Zhang, M. Qian, O. Faruque, and M. Brandt, “Slm lattice structures: Properties, performance, applications and challenges,” *Materials Design*, vol. 183, p. 108137, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0264127519305751

[39] C. T. Richard and T.-H. Kwok, “Analysis and design of lattice structures for rapid-investment casting,” *Materials*, vol. 14, no. 17, p. 4867, 2021.

[40] H. Proff and A. Staffen, “Deloitte challenges of additive manufacturing,” Feb 2019. [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/de/Documents/operations/Deloitte_Challenges_of_Additive_Manufacturing.pdf

[41] D. Dimitrov, E. Uheida, G. Oosthuizen, D. Blaine, R. Laubscher, A. Sterzing, P. Blau, W. Gerber, and O. Damm, “Manufacturing of high added value titanium components. a south african perspective,” *IOP Conference Series: Materials Science and Engineering*, vol. 430, p. 012009, 10 2018.

[42] A. Pearson, “Airbus standardizes on stratasys additive manufacturing solutions for aircraft supply chain.” [Online]. Available: https://www.stratasys.com/explore/blog/2016/airbus-additive-manufacturing-ultem-9085

[43] A. Bentur, “The am applications series: Antennas.” [Online]. Available: http://www.leolane.com/blog/applications-series-antennas/

[44] M. Pástor, M. Binda, and T. Harčarik, “Modal assurance criterion,” *Procedia Engineering*, vol. 48, p. 543–548, 12 2012.

[45] T. D. Tsai and C. C. Cheng, “Structural design for desired eigenfrequencies and mode shapes using topology optimization,” *Struct. Multidiscip. Optim.*, vol. 47, no. 5, p. 673–686, May 2013. [Online]. Available: https://doi.org/10.1007/s00158-012-0840-2

[46] D. Ruiz, J. Bellido, and A. Donoso, “Eigenvector sensitivity when tracking modes with repeated eigenvalues,” *Computer Methods in Applied Mechanics and Engineering*, vol. 326, pp. 338–357, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045782517305728

[47] Q. Xu, E. Wehrle, and H. Baier, *Adaptive and Engineering Knowledge based Metamodeling in Multidisciplinary Design Optimization of Aircraft Wing Structures*. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2012-5448

[48] A. Petras, “Design of sandwich structures,” Ph.D. dissertation, University of Cambridge, 1999.

[49] T. Khan, V. Acar, and Aydin, “A review on recent advances in sandwich structures based on polyurethane foam cores,” *Polymer Composites*, vol. 41, no. 6, pp. 2355–2400, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/pc.25543

A

Code of Project

Listing A.1: Dailey's method.m

```
function [DU,dMAC,Diele_full_patch, Diele_patch]=Daileys_method_dMAC(x,wns,Diele,dcM_elw)

global NW ND CellL StiffnessCand Pen NuMod_ob Diele_Full
global Cell MassCand NumNat FilterON Yc
global SolidEleOr


n=ND*NW;
[w,dw,ddw]=Weight_2M(x,Pen,0);
[wM,dwM,ddwM]=Weight_2M(x,1,1);
DU=[];
```

```
%%
for ifN=1:NuMod_ob
    g_Cons_T=0;
    for cl=1:ND
            CelAco=[];
            CelAco=Cell{cl};
            MAC_T=0;
            dMac_T=0;
            for i=1:length(CelAco)
                CEnu=CelAco(i);
                CEnu=CelAco(i);
                 MAC_Cons_tm=(((Diele(:,CEnu,ifN)'*Diele_Full(:,CEnu,ifN))^2)./...
                     ((Diele(:,CEnu,ifN)'*Diele(:,CEnu,ifN))*...
                     (Diele_Full(:,CEnu,ifN)'*Diele_Full(:,CEnu,ifN))))-1;
                 MAC_T=MAC_T+MAC_Cons_tm;
            end
            g_Cons_T=g_Cons_T+MAC_T;
    end
    g_Cons(:,ifN)=g_Cons_T;
end
Eig_vec_C=g_Cons;

[Rep_pos]=EV_reordering(wns);

for ifN=1:NuMod_ob      %eigenmodes
    for cl=1:ND         %patches
        for j=1:NW      %materials
            kci=0;
            CelAco=[];
            CelAco=Cell{cl};
            DUt=zeros(24,1);
            %MAC_T=0;
            dMac_T=0;
            for i=1:length(CelAco) %elements
                if  Rep_pos(ifN)==0
                    CEnu=CelAco(i);
                    G_t_bar=[];
                    f_t_bar=[];
                    V=[];
                    c=0;
                    Uj=Diele(:,CEnu,ifN);
                    D_day = Uj'*(dw(j,cl)*StiffnessCand(:,:,CEnu,j)-...
                        ((wns(ifN))*dwM(j,cl)*MassCand(:,:,CEnu,j)))*Uj;
                    [lambda,gamma]=eig(D_day);

                    Z_day=Uj*lambda;
                    G_t_bar(:,:)=w(j,cl)*StiffnessCand(:,:,CEnu,j)-...
                        ((wns(ifN))*wM(j,cl)*MassCand(:,:,CEnu,j));
                    f_t_bar(:,1)=dcM_elw(j,CEnu,cl,ifN)*wM(j,cl)*...
                        MassCand(:,:,CEnu,j)*Z_day-((dw(j,cl)*...
                        StiffnessCand(:,:,CEnu,j)-abs(((wns(ifN)))*...
                        dwM(j,cl)*MassCand(:,:,CEnu,j))))*Z_day;

                    ntm=max(Z_day);
                    [xx,yy]=find(Z_day==ntm);

                    G_t_bar(:,xx)=0; G_t_bar(xx,:)=0; f_t_bar(xx,1)=0;

                    for ii=1:24
                        G_t_bar(ii,ii)=1;
                    end

                    V_day=G_t_bar\f_t_bar;

                    Q_day=V_day'*wM(j,cl)*MassCand(:,:,CEnu,j)*Z_day-...
                        Z_day'*wM(j,cl)*MassCand(:,:,CEnu,j)*V_day-Z_day'*...
                        dwM(j,cl)*MassCand(:,:,CEnu,j)*Z_day;

                    R_day_1=Z_day'*(w(j,cl)*StiffnessCand(:,:,CEnu,j)-...
                        ((wns(ifN))*wM(j,cl)*MassCand(:,:,CEnu,j)))*V_day;
                    R_day_2=-Z_day'*(dwM(j,cl)*MassCand(:,:,CEnu,j)*...
                        Z_day-wM(j,cl)*MassCand(:,:,CEnu,j)*V_day);
                    R_day_3=0.5*Z_day'*(ddw(j,cl)*StiffnessCand(:,:,CEnu,j)...
                        -((wns(ifN))*ddwM(j,cl)*MassCand(:,:,CEnu,j)))*Z_day;

                    R_day=R_day_1+R_day_2+R_day_3;

                    C_day= 0.5 * Q_day;

                    EV_der= V_day + Z_day*C_day;

                    DUt=DUt+EV_der;
                else
                    if Rep_pos(ifN)==1
                        KK=find(Rep_pos==1);
                        n_rep=size(KK,2);
                    elseif Rep_pos(ifN)==2
                        KK=find(Rep_pos==2);
                        n_rep=size(KK,2);
                    elseif Rep_pos(ifN)==3
                        KK=find(Rep_pos==3);
                        n_rep=size(KK,2);
                    end
                    CEnu=CelAco(i);
                    G_t_bar=[];
                    f_t_bar=[];
                    V=[];
                    c=0;
                    Uj=[];
                    Ujj=[];
                    CEnu=CelAco(i);
                    for ik=1:n_rep
                        Ujj(:,ik)=Diele(:,CEnu,ik);
                        Uj(:,ik)=Ujj(:,ik);
                    end
                    D_day = Uj'*(dw(j,cl)*StiffnessCand(:,:,CEnu,j)-...
                        ((wns(ifN))*dwM(j,cl)*MassCand(:,:,CEnu,j)))*Uj;
                    [lambda,gamma]=eig(D_day);

                    Z_day=Uj*lambda;
                    G_t_bar(:,:)=w(j,cl)*StiffnessCand(:,:,CEnu,j)-...
                        ((wns(ifN))*wM(j,cl)*MassCand(:,:,CEnu,j));

                    for i=1:n_rep
                        f_t_bar(:,i)=gamma*wM(j,cl)*MassCand(:,:,CEnu,j)*...
                            Z_day(:,i)-((dw(j,cl)*StiffnessCand(:,:,CEnu,j)-...
                            abs(((wns(ifN)))*dwM(j,cl)*...
                            MassCand(:,:,CEnu,j))))*Z_day(:,i);
                    end
                    ntm=max(Z_day);
                    [xx,yy]=find(Z_day==ntm);

                    G_t_bar(:,xx)=0; G_t_bar(xx,:)=0; f_t_bar(xx,:)=0;

                    for ii=1:24
                        G_t_bar(ii,ii)=1;
                    end

                    V_day=G_t_bar\f_t_bar;

                    Q_day=V_day'*wM(j,cl)*MassCand(:,:,CEnu,j)*Z_day-...
                        Z_day'*wM(j,cl)*MassCand(:,:,CEnu,j)*V_day-Z_day'
```

```
                        ...
                    *dwM(j,cl)*MassCand(:,:,CEnu,j)*Z_day;

                R_day_1=Z_day'*(w(j,cl)*StiffnessCand(:,:,CEnu,j)...
                -((wns(ifN))*wM(j,cl)*MassCand(:,:,CEnu,j)))*V_day;
                R_day_2=-Z_day'*(dwM(j,cl)*MassCand(:,:,CEnu,j)...
                    *Z_day-wM(j,cl)*MassCand(:,:,CEnu,j)*V_day);
                R_day_3=0.5*Z_day'*(ddw(j,cl)*StiffnessCand(:,:,CEnu,j)...
                    -((wns(ifN))*ddwM(j,cl)*MassCand(:,:,CEnu,j)))*Z_day;

                R_day=R_day_1+R_day_2+R_day_3;

                for ik=1:n_rep
                    kk=gamma(ik,ik);
                    C_day=R_day/kk;
                end

                EV_der= V_day + Z_day * C_day;

                DUt=DUt+EV_der;
            end
        end

        if size(DUt,2)==1
            DU(:,j,cl,ifN) = DUt;
        else
            for i=1:n_rep
                DU(:,j,cl,ifN) = DUt(:,i);
            end
        end

    end
end
```

```

end

%Mac Derivatives Calculation

%MACDERIVATIVES
Diele_full_patch=zeros(24,ND,NuMod_ob);
Diele_patch=zeros(24,ND,NuMod_ob);

for m=1:NW
   for j=1:ND
      for k=1:NuMod_ob
          CelAco=[];
          CelAco=Cell{j};
          UF=zeros(24,1);
          US=zeros(24,1);
          for i=1:length(CelAco)
                  CEnu=CelAco(i);
                  UF=UF+Diele_Full(:,CEnu,k);
                  US=US+Diele(:,CEnu,k);
          end
          Diele_full_patch(:,j,k)=UF;
          Diele_patch(:,j,k)=US;

      end
   end
end

a=Diele_full_patch;
b=Diele_patch;

Squ=zeros(24,ND,NuMod_ob);
for i=1:NuMod_ob
    for j=1:ND
        Squ(:,j,i)=(((2*(a(:,j,i)'*b(:,j,i))/((a(:,j,i)'*a(:,j,i))*...
            (b(:,j,i)'*b(:,j,i))))*(a(:,j,i))'+(((2*(a(:,j,i)'*...
            b(:,j,i))^2))/((a(:,j,i)'*a(:,j,i))*((b(:,j,i)'*...
            b(:,j,i))^2)))*(b(:,j,i)')));
    end
end

dMAC=zeros(NW,ND,NuMod_ob);

for i = 1:NuMod_ob
    for j = 1:NW
        for k = 1:ND
            dMAC(j,k,i) = Squ(:,k,i)' * DU(:,j,k,i);
        end
    end
end

%dMAC=reshape(dMAC,[NW*ND,NuMod_ob]);

if FilterON==1

    for ifN=1:NuMod_ob
        DUi=dg_Cons(:,ifN);
        DEig_vec_C(:,ifN)=YFiltering(DUi,ND);
    end

end
```

Listing A.2: `Adjoint Method.m`

```
function [dMAC,Diele_PATCH,Diele_full_PATCH]=Adj_met(wns, x)

global Diele_UN_full Diele_UN StiffnessCand MassCand NuMod_ob Pen NW ND Cell

[M,K]=Global_stiffness;
GlobalMASS=M;
GlobalSTIFF=K;

Diele_obj_T=Diele_UN_full;
Diele_T=Diele_UN;

for i=1:NuMod_ob

    k=((2*Diele_T(:,i)'*Diele_obj_T(:,i))/((Diele_T(:,i)'*Diele_T(:,i))*(Diele_obj_T(:,i)'*Diele_T(:,i))))*Diele_obj_T(:,i)';
    j=(2*(Diele_obj_T(:,i)'*Diele_T(:,i))^2/((Diele_obj_T(:,i)'*Diele_obj_T(:,i))*(Diele_T(:,i)'*Diele_T(:,i))^2)*Diele_T(:,i)');
    a=(k-j)*Diele_T(:,i);
    a_j(i)=a;

    delta = GlobalSTIFF - wns(i)*GlobalMASS;
    c = (k-j) - 2*a*Diele_T(:,i)'*GlobalMASS;

    alpha_p = delta'\c';

    b = -alpha_p'*GlobalMASS*Diele_T(:,i);

    alpha_j(:,i) = b * Diele_T(:,i) + alpha_p;
end

ENode=dlmread('C:/Temp/Conectivity.txt');

for ina=1:NuMod_ob
```

```
    U=alpha_j(:,ina);
    Di=Diele_T(:,ina);
    Df=Diele_obj_T(:,ina);

    for ii=1:19200  %Number of Elements
        for j=1:24
            CEN=ENode(ii,:);
            if mod(j/3,1) == 0
                kii=3*CEN(j/3);
                Ued(j)=U(kii);
                Die(j)=Di(kii);
                DieF(j)=Df(kii);
            elseif mod(((j+1)/3),1) == 0
                ki=3*CEN((j+1)/3)-1;
                Ued(j)=U(ki);
                Die(j)=Di(ki);
                DieF(j)=Df(ki);
            else
                ki=3*CEN((j+2)/3)-2;
                Ued(j)=U(ki);
                Die(j)=Di(ki);
                DieF(j)=Df(ki);
            end
        end
        alpha_j_ELEM(:,ii,ina)=Ued';
        Diele_ELEM(:,ii,ina)=Die';
        Diele_full_ELEM(:,ii,ina)=DieF';
    end
end

[w,dw,ddw]=Weight_2M(x,Pen,0);
[wM,dwM,ddwM]=Weight_2M(x,1,1);

for j=1:ND
```

```
   for k=1:NuMod_ob
      CelAco=[];
      CelAco=Cell{j};
      UF=zeros(24,1);
      DD=zeros(24,1);
      DDF=zeros(24,1);
      for i=1:length(CelAco)
          CEnu=CelAco(i);
          UF=UF+alpha_j_ELEM(:,CEnu,k);
          DD=DD+Diele_ELEM(:,CEnu,k);
          DDF=DDF+Diele_full_ELEM(:,CEnu,k);
      end
      alpha_j_PATCH(:,j,k)=UF;
      Diele_PATCH(:,j,k)=DD;
      Diele_full_PATCH(:,j,k)=DDF;
   end
end

KK=zeros(24,24);
MM=zeros(24,24);

for j=1:ND
    for k=1:3
      CelAco=[];
      CelAco=Cell{j};
      UF=zeros(24,1);
      for i=1:length(CelAco)
           CEnu=CelAco(i);
           KK=KK+StiffnessCand(:,:,CEnu,k);
           MM=MM+MassCand(:,:,CEnu,k);
      end
      StiffnessCand_PATCH(:,:,j,k)=KK;
      MassCand_PATCH(:,:,j,k)=MM;
    end
end
```

```

for ifN=1:NuMod_ob      %eigenmodes
    for cl=1:ND         %patches
        for j=1:NW      %materials
            dMAC(j,cl,ifN) = alpha_j_PATCH(:,cl,ifN)'*(dw(j,cl)*StiffnessCand(:,:,cl,j)*Diele_PATCH(:,cl,ifN))...
                +(((a_j(ifN)*Diele_PATCH(:,cl,ifN)'-wns(ifN)*alpha_j_PATCH(:,cl,ifN)'))*(dwM(j,cl)*MassCand(:,:,cl,j)*Diele_PATCH(:,cl,ifN)))';
        end
    end
end
```