LAMINATION PARAMETER OPTIMIZATION OF FLAT FIBRE REINFORCED PLATES FOR VIBRATION FREQUENCY CRITERIA

Miguel Ferreira Baltazar de Matos Almas

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Examination Committee

Chairperson: Prof. Luís Manuel Varejão Oliveira Faria
Supervisor: Prof. José Arnaldo Pereira Leite Miranda Guedes
Co-Supervisor: Prof. Hélder Carriço Rodrigues
Members of the Committee: Prof. Aurélio Lima Araújo

November of 2013
Acknowledgments

First and foremost, I would like to express my gratitude to Professors José Miranda Guedes and Hélder Rodrigues for their advice, support and guidance when developing this work. Their vast knowledge of these subjects was essential in greatly expanding my own understanding of the field of discrete optimization.

Secondly I would like to acknowledge professor José Herskovits who wrote the optimizer used in this thesis named Feasible Arc Interior Point Analysis (FAIPA), as well as Professor Aurélio Araújo for his insights as to how FAIPA functioned. I would also like to acknowledge the contributions of all the authors of the quoted references, as these constitute the foundations on which this work is built upon.

Last but not least I would like to thank my family for all the support and encouragement given.
Abstract

One of the main challenges faced in the design of continuous fibre reinforced composite laminates is the optimal orientation and stacking order. The work was developed with the aim of developing an application that will optimize a multi-layered flat plate, with regards to its fibre orientation for vibration criteria. Fibre reinforced composite material design guidelines stipulate that only a handful of fibre orientations should be chosen which means that the optimization problem is discrete.

To this end, the Discrete Material Optimization (DMO) method was applied and the application was implemented with Matlab and Ansys. As the total mass of fibre and matrix materials are kept constant, the optimization is simplified due to the inexistence of mass constraints. By keeping the optimization problem simple, the effectiveness of the DMO method can be more accurately measured and its shortcomings identified.

Several examples are presented, solved and analysed in this work. These results show that the application can obtain optimal results at a fraction of the time required by an algorithm that tests all possible stacking orders, the computation time being at least 3 and 10 times faster in 4 and 5 layered cases respectively. Furthermore, deficiencies of the DMO and problem formulation were identified that once overcome can help the development of future, more capable optimizing methodologies.

Keywords: Vibration Analysis, Discrete Material Optimization (DMO), Continuous Fibre Reinforced Composite Materials, Fibre Orientation, Stacking Order.
Resumo

Um dos desafios que o projecto de componentes em materiais compósitos reforçados por fibra enfrenta é a obtenção da ordem ideal de empilhamento. O trabalho desenvolvido tem como objectivo principal o desenvolvimento de uma aplicação que optimize uma placa plana com múltiplas camadas, para critérios de vibração. As directrizes do projecto de materiais compósitos reforçados por fibra estipulam que apenas determinadas orientações de fibras devem ser utilizadas pelo que o problema de optimização é discreto.

Para este fim, foi utilizado o método DMO (Discrete Material Optimization) e a aplicação foi implementada recorrendo aos programas Matlab e Ansys. Como os materiais da fibra e da matriz são mantidos constantes, o modelo de optimização é simplificado pela eliminação da necessidade de constrangimentos de massa. Esta simplicidade permite uma melhor avaliação do comportamento não só do programa, mas do próprio método DMO em que ele assenta.

Neste trabalho são apresentados, resolvidos e analisados vários exemplos. Estes comprovam que a aplicação consegue obter resultados óptimos, numa fracção do tempo necessário para analisar todas as possíveis combinações de orientação de fibra, com tempos 3 vezes inferiores em casos com 4 lâminas e mais de 10 vezes menores em casos com 5 lâminas. Além disso, falhas método DMO e da formulação do problema foram identificadas e quando corrigidas deverão possibilitar o desenvolvimento no futuro, de melhores e mais capazes metodologias de optimização.

Palavras-chave: Análise à Vibração, Discrete Material Optimization (DMO), Materiais Compósitos Reforçados por Fibra Contínua, Orientação de Fibra, Ordem de Empilhamento.
# Table of Contents

1. Introduction  
   1.1. Composite Materials: A brief historical perspective  
   1.2. Vibrations – Importance and Impact  
   1.3. Objective and Motivations  
      1.3.1. Practical Considerations  

2. State of the Art  
   2.1. Topology Optimization  
   2.2. DMO (Discrete Material Optimization)  
      2.2.1. DMO Scheme 4 (DMO4)  
      2.2.2. DMO Scheme 5 (DMO5)  
   2.3. Finite Element Analysis (FEA)  

3. Optimization  
   3.1. Problem Formulation  
   3.2. Sensitivity Analysis  
      3.2.1. Eigenvalue Problem  
      3.2.2. Lagrangian and Sensitivity Analysis  
   3.3. Brief description of the optimizer used  
      3.3.1. Quasi-Newton method  
      3.3.2. Line Search – Armijo, Wolfe and Goldstein Conditions  

4. Composite Materials  
   4.1. Material properties  
   4.2. Manufacturing Technologies  
      4.2.1. Wet Lay-up/Hand Lay-up  
      4.2.2. Vacuum bagging  
      4.2.3. Filament-winding  
      4.2.4. Compression Moulding  
      4.2.5. Pultrusion  
      4.2.6. Resin Transfer Moulding (RTM)  

5. Computational model  
   5.1. Plate Theory
5.1.1. ESL – Classical Laminate Plate Theory (CLPT)  
5.1.2. First-Order Shear Deformation Theory (FSDT)  
5.2. Finite Elements  
5.3. Penalty Selection  
5.4. Mode of Vibration Superposition and Switching  
5.4.1. Mode Superposition  
5.4.2. Mode Switching  
5.5. Algorithm  
5.5.1. Program's intended functionalities  
5.5.2. Overview of the program's structure  
5.5.3. Step by Step Description of the Program  
6. Testing and Analysis of the Results  
6.1. Material Properties  
6.2. Test Case 1  
6.3. Test Case 2  
6.4. Test Case 3  
6.5. Test Case 4  
6.6. Test Case 5  
7. Conclusions and Future Work  
8. References  
A. Additional Test Cases  
A.1. Test Case 6  
A.2. Test Case 7  
A.3. Test Case 8  
A.4. Test Case 9  
A.5. Test Case 10  
A.6. Test Case 11  
B. User’s Manual  
B.1. Creating the Problem Instance  
B.2. Solving  
B.3. Post-Processing
List of Figures

Figure 1-1 – 1953 Chevrolet Corvette. Picture from http://www.autoweek.com/article/20130628/carnews/130629840 ......................................................... 1
Figure 1-2 – The Beechcraft Starship. Picture found in http://en.wikipedia.org/wiki/Beechcraft_Starship .......................................................... 2
Figure 1-3 – The Tacoma Narrows Bridge Resonating. Image obtained from http://www.fhwa.dot.gov/publications/publicroads/11janfeb/03.cfm ........................................... 3
Figure 1-4 – Representation of the recommended fibre orientation angles. 1) 0º; 2) 90º; 3) 45º; 4) -45º. ........................................................................................................ 4
Figure 2-1 – (Above) Design domain. (Below) Solution obtained from topology optimization ...... 7
Figure 2-2 – Image taken from [6] which represents the sum of two variables w1 and w2, for two materials, computed with DMO scheme 4. .................................................................... 9
Figure 2-3 – Picture found in [6] which represents weight functions w1 and w2 for two materials computed with DMO scheme 5. ......................................................................... 11
Figure 3-1 – Image taken from [23]; Steepest Descent Steps .......................................................... 17
Figure 3-2 – Image taken from [23] – The ideal step length is the optimizer .......................... 18
Figure 3-3 – Image taken from [23]: Sufficient decrease or Armijo condition .................... 19
Figure 4-1 – Fibre reinforced composite material with the resin in blue and the substrate in yellow. Image from http://en.wikipedia.org/wiki/File:Composite_3d.png ........................................ 21
Figure 4-2 - Materials used in the Boeing 787 Dreamliner. Image found in http://bintang.site11.com/Boeing_787/Boeing787_files/Specifications.html ........................................ 22
Figure 4-3 – Wet Lay-up production technique. Image from [26]. ............................................. 23
Figure 4-4 – Vacuum bagging manufacturing technique. Image from [26]. ................................... 24
Figure 4-5 – Filament winding process. Image from [26]. ....................................................... 24
Figure 4-6 – Compression moulding method. Image from [26]. .............................................. 25
Figure 4-7 – Pultrusion production technique. Image from [26]. ............................................. 25
Figure 4-8 – Resin transfer moulding. Image from [26]. ...................................................... 26
Figure 5-1 - Undeformed and deformed geometry under the Classical Laminate Theory. Picture from [27]. ............................................................................................................. 28
Figure 5-2 – Deformed and Undeformed geometry under the assumptions of the First Order Shear Deformation Theory. Image from [27]. ................................................ 29
Figure 5-3 – Axysimmetric geometry modelled with Ansys. Image from http://www.mece.ualberta.ca/tutorials/ansys/IT/Axysymmetric/Axysymmetric.html ............... 30
Figure 5-4 – Shear locking of a linear element. On the left is the realistic deformation and on the right the one obtained with these elements. Image found in http://www.sintef.no/project/eVITameeting/2012/KMM%20Geilo%202012%20Lecture%2010.pdf ........................................................................................................ 31
Figure B-6 – Txt.m commands for the input of the degrees of freedom that are constrained. ... 78
Figure B-7 – Lines controlling the output printing to the txt file with all Ansys commands. In this
  case only 1 line is constrained, dl1................................................................. 79
Figure B-8 – Txt_aux.m commands for the input of the degrees of freedom that are constrained.
  ................................................................................................................. 79
Figure B-9 - Lines controlling the output printing to the txt_aux file with all Ansys commands. In
  this case only 1 line is constrained, dl1 .......................................................... 79
Figure B-10 – Penalty scheme in Fun.m file. .............................................................. 80
Figure B-11 – Derivatives rescaling. ........................................................................ 80
Figure B-12 – Line of Inicial.m responsible for firing up the post-processor..................... 80
Figure B-13 – Example of the feedback obtained from the optimizer ............................. 81
Figure B-14 – Total number of fibre possibilities in post-processing on a 5 layered case........ 82
List of Tables

Table 1 - Material properties for the test cases ........................................................................... 41
Table 2 – Main results of test case 1 ............................................................................................... 42
Table 3 – Weight functions’ values for test case 1 ........................................................................ 44
Table 4 – Main results of test case 2 ............................................................................................... 45
Table 5 - Weight functions’ values for test case 2 ........................................................................ 46
Table 6 – Difference between orientations in layer 3 .................................................................... 47
Table 7 – Design variables and derivatives on the third iteration .................................................... 48
Table 8 - Main results of test case 3 ............................................................................................... 50
Table 9 - Weight functions’ values for test case 3 .......................................................................... 51
Table 10 - Main results of test case 4 ............................................................................................. 53
Table 11 - Weight functions’ values for test case 4 ....................................................................... 54
Table 12 – Difference between the natural frequencies obtained with the stacking orders of the
global maximum and the local maximum obtained by the optimizer ............................................ 55
Table 13 - Main results of test case 5 ............................................................................................. 57
Table 14 – First and second frequencies on the first and last iterations of optimization ................. 58
Table 15 – Derivative’s values on the first two iterations of problem 5 with 5 layers ..................... 60
Table 16 - Weight functions’ values for test case 5 ....................................................................... 61
Table 17 – Main results of test case 6 ............................................................................................. 69
Table 18 - Main results of test case 7 ............................................................................................. 70
Table 19 - Main results of test case 8 ............................................................................................. 71
Table 20 - Main results of test case 9 ............................................................................................. 72
Table 21 - Main results of test case 10 ............................................................................................ 73
Table 22 - Main results of test case 11 ............................................................................................ 74
<table>
<thead>
<tr>
<th><strong>Acronyms</strong></th>
<th><strong>Definition</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ANS</td>
<td>Assumed natural strain</td>
</tr>
<tr>
<td>BFGS</td>
<td>Broyden-Fletcher-Goldfarb-Shanno</td>
</tr>
<tr>
<td>CLPT</td>
<td>Classical Laminated Plate Theory</td>
</tr>
<tr>
<td>DFP</td>
<td>Davidon-Fletcher-Powell</td>
</tr>
<tr>
<td>DMO</td>
<td>Discrete Material Optimization</td>
</tr>
<tr>
<td>ESL</td>
<td>Equivalent single layer</td>
</tr>
<tr>
<td>FAIPA</td>
<td>Feasible Arc Interior Point Analysis</td>
</tr>
<tr>
<td>FDIPA</td>
<td>Feasible Direction Interior Point Analysis</td>
</tr>
<tr>
<td>FEA</td>
<td>Finite Element Analysis</td>
</tr>
<tr>
<td>FSDT</td>
<td>First order shear deformation theory</td>
</tr>
<tr>
<td>HB</td>
<td>Harwell – Boeing</td>
</tr>
<tr>
<td>KKT</td>
<td>Karush-Kuhn-Tucker</td>
</tr>
<tr>
<td>MMA</td>
<td>Method of Moving Asymptotes</td>
</tr>
<tr>
<td>RTM</td>
<td>Resin transfer moulding</td>
</tr>
<tr>
<td>SIMP</td>
<td>Solid Isotropic with Material Penalization</td>
</tr>
<tr>
<td>3-D</td>
<td>Three-Dimensional</td>
</tr>
</tbody>
</table>
## Symbols

- $a$ Plate length
- $B_k$ Hessian matrix
- $C^l$ Single layer constitutive matrix
- $C_i$ Fibre oriented constitutive matrix (Orientation defined by $i$).
- $[c]$ Damping matrix
- $[C]$ Stress-strain matrix
- $C_{ij}$ Stress-strain matrix coefficients
- $[D]$ *Ansys* anisotropic stress-strain matrix
- $D$ Bending stiffness
- $d_k$ Search direction
- $E$ Young's modulus
- $\{F\}$ Force vector
- $f$ Objective function
- $G$ Shear modulus
- $h$ Plate thickness
- $[K]$ Stiffness matrix
- $L$ Lagrangian
- $[M]$ Mass matrix
- $p$ Penalty
- $[T]$ Matrix of coordinate transformation
- $[\tilde{T}]$ *Ansys* matrix of coordinate transformation
- $U_{1i}, U_{2i}, \ldots$ Lagrange multiplier
- $U_{3mn}$ Mode shape
- $U_0$ Displacement along the $x$ direction
- $V_0$ Displacement along the $y$ direction
$W_o$ Displacement along the $z$ direction

$w_i$ Weight variable

$\bar{\omega}_i$ Weight function

$x_i$ Design variable

$x_i^-$ Lower bound constraint of $x_i$

$x_i^+$ Upper bound constraint of $x_i$

$\{x\}$ Displacement vector

$\{\dot{x}\}$ Velocity vector

$\{\ddot{x}\}$ Acceleration vector

$\{X\}$ Eigenvector

Greek Symbols

$\lambda$ Fundamental frequency

$\nu$ Poisson’s ratio

$\rho$ Density

$\phi$ Eigenvector

$\bar{\phi}$ Mass normalized eigenvector

$\phi_x$ Transverse normal rotation along $y$

$\phi_y$ Transverse normal rotation along $y$

$\omega, \omega_{mn}$ Natural Frequency
1. Introduction

Throughout this chapter a brief overview of the subjects of this work will be presented, as well as an explanation of its objectives.

1.1. Composite Materials: A brief historical perspective

Even though composite materials have been known and used throughout humankind’s history [1], their modern history starts only in the 20th century with the commercialization of fiberglass by the Owens Corning Fiberglass Company in the year of 1937. Since then, new materials with increasingly superior and more varied mechanical properties have been developed and used.

During the Second World War several breakthroughs were achieved, such as the first fiberglass hull boat (1942) and the first structural use of composite materials in aircraft design on some experimental At-6 and BT-15 wings (1942). Despite the successful flight of these aircrafts, no further attempts to use composite materials on aircraft structural components would be undertaken for around 50 years. Other notable breakthroughs of this period include the development of spray layup and filament winding production techniques.

With the end of the war, the manufacturers began looking for civilian applications to these technologies, notably in the naval sector with the production of commercial fiberglass boat hulls, as well as in the automotive industry. A notable example of this was the 1953 Corvette whose body was made from fiberglass reinforced plastic. Other products made from composite materials include tubs, non-corrosive pipes and even furniture [2].

Figure 1-1 – 1953 Chevrolet Corvette. Picture from http://www.autoweek.com/article/20130628/carnews/130629840
The aerospace industry has always been a great source of innovation in the composite materials field. During the 60’s many advances were made on the filament winding technique resulting in a greater production precision. It was also the decade which saw the invention of the first quality carbon fibre, first by A. Shindo in 1961 and later in the United Kingdom in 1963 where the high strength potential of the fibre was achieved. Spurred by this development, by 1968 Rolls-Royce was building the first engine rotors which would prove disastrous as these were shown to be fragile after failing the so-called “bird collision test”. Another important milestone was the start of the development of the Beechcraft Starship in 1979 [3] since it would have the first fully filament wound aircraft fuselage.

![Figure 1-2 – The Beechcraft Starship. Picture found in http://en.wikipedia.org/wiki/Beechcraft_Starship](http://en.wikipedia.org/wiki/Beechcraft_Starship)

Nowadays, the composite material market continues its rise, as the technology is developed, quality increases and price decreases, allowing them to be used in more applications than ever before.

### 1.2. Vibrations – Importance and Impact

Vibration is the oscillatory response of a mechanical system and can be either free or externally forced. Free vibrations occur without applied loads and are representative of the natural dynamic vibration of the mechanical system. Forced vibrations happen due to external action (excitation force) [4].

Mechanical systems have natural frequencies of vibration which are also called resonance frequencies. If a periodic force acts on the system at the natural frequency then it will experience oscillations with increasing amplitudes that will far exceed those verified at other forced frequencies. This phenomenon is called forced resonance and represents a significant risk as it can cause catastrophic mechanical damage. The classical example of this is the Tacoma Narrows Bridge [5] which collapsed on the 7th of November of 1940 due to action of the...
wind which caused aero elastic flutter effects on the bridge in turn causing great displacements until the point of collapse.

![Image](http://www.fhwa.dot.gov/publications/publicroads/11janfeb/03.cfm)

**Figure 1-3 – The Tacoma Narrows Bridge Resonating.** Image obtained from http://www.fhwa.dot.gov/publications/publicroads/11janfeb/03.cfm

Vibrations can also impact the lifetime of a mechanical component by causing fatigue damage due to repeating oscillation. An airplane wing subject to such effects could in time collapse due to accumulated fatigue damage. Vibration can also loosen fasteners such as bolts or cause mechanical noise [5].

Vibrations can also have a direct impact on the feeling of comfort (or absence of it) on the user of a mechanical system and can in extreme cases even cause health issues. Consider the vibrations of an airplane; if its frequency of vibration is equal to the natural frequency of the human intestinal tract (from 4 to 8 Hz), serious internal damage can occur on the passengers [4].

It is therefore clear that some care should be given to the frequency response of a system in order to ensure its correct operation. A solution can be to ensure that the natural frequency of a system only occurs above its operational frequencies. This can be achieved by a reduction of mass, increase of its stiffness or a combination of both.

### 1.3. Objective and Motivations

When designing a component in fibre reinforced layered composite material one is confronted with a question: What orientation should the fibres have in each layer? Although an engineer’s intuition can yield a good configuration, in complex cases it won’t probably be the optimal solution.
There is therefore an interest in developing an optimization program for a multi-layered fibre reinforced composite material plate with the goal of obtaining the best orientation configuration to ensure maximum (or minimum) fundamental frequency.

On the literature one will find that indeed such a program has already been developed in [6], [7], [8] and [9]. It was, however, developed using academic analysis tools. The primary goal of this work is to create such a tool with commercially available programs.

*Matlab* programming language was used for writing the code and *Ansys* for finite element calculations required for obtaining the stiffness and mass matrixes needed for the eigenvalue and eigenvector calculations.

This work also aims at investigating the performance of the DMO method described in [6] and to identify both its main advantages and shortcomings.

### 1.3.1. Practical Considerations

When developing an application, one must consider practical aspects that will ensure it has the widest possible scope of application.

![Figure 1-4 – Representation of the recommended fibre orientation angles. 1) 0°; 2) 90°; 3) 45°; 4) -45°.](image)

If one described the problem as a continuous one, the final fibre orientations could take any value contained in the [-90°, 90°] interval. The orientations obtained from this interval might indeed be a mathematically superior choice, but it would also almost certainly be commercially unviable to produce. This is because industrially, only certain specific orientations are usually selected such as [0°, 90°, 45°, -45°] [10]. By forcing the optimizer to select one of these
orientations, unnecessary production costs can be avoided. The optimization problem is therefore a discrete one.

However, by using the DMO method this discrete problem is reformulated as a continuous one, allowing the use of continuous optimization methods for its solution. The application should therefore choose the best fibre orientation, for each ply, from the set of angles defined by the user.
2. State of the Art

This chapter is dedicated to the description of the latest developments on the fields of structural optimization and shell finite elements for composite materials.

2.1. Topology Optimization

As stated earlier in section 1.3, the optimization problem object of this work is naturally a discrete problem. The models used to overcome this difficulty, are closely related to the ones developed to overcome similar difficulties in the field of topology optimization.

This field of optimization started approximately 30 years ago and its objective is the definition of the layout of a structure within a given admissible domain, with respect to a given cost function and design constraints. It is therefore a discrete optimization problem whose variables vary between 0 and 1. Its modern basis was laid out in Bendsøe and Kikuchi (1988), [11] who used the homogenization technique and by Bendsøe (1989) [12] who introduced the Solid Isotropic Material with Penalization (SIMP) method to relax the problem into a continuous form approximating the original discrete problem. Since then many developments have taken place.

Thus topology optimization is used to obtain the best possible geometry of a structure that is subject to certain loads and design constraints. It does this by assigning a continuous design variable (a “density” varying between 0 and 1) for each finite element on the design domain. These variables will then be pushed to the limit values by a penalisation method such as SIMP and form a picture made of elements in greyscale, where white elements have 0 density and are therefore empty (to avoid numerical problems, a lower limit is normally set such as 0.001) and black elements have their variables set at 1 and represent areas where material should be applied (see Figure 2-1).

Even though this problem is not the one we’re concerned with, it is close enough to have spawned a new theory that tries to solve the problem of fibre orientation in composite materials. This theory is named DMO and was developed by Jan Stegmann on his PHD thesis [6].

![Figure 2-1 – (Above) Design domain. (Below) Solution obtained from topology optimization](image-url)
2.2. DMO (Discrete Material Optimization)

DMO was developed to solve problems where both the composite material’s fibre orientation and the material itself are simultaneously optimized, for any given number of layers, materials, or angles of orientation.

This method works in a similar fashion to the topology optimization method (SIMP Model) described in chapter 2.1. The main difference is that now, each of the design variables \( x_i \) represents one of the possible orientations, on each layer. These should still be pushed to their extreme values of either 0, in which case it is not an ideal orientation, or 1 if it is. Since there can be any given number of possible fibre orientations per layer and that only one of these should have unitary value, they must all be related to each other mathematically. This means that a simple SIMP method can no longer be used and new interpolation equations must be used. These are going to be described in detail later on.

For now consider that to obtain the layer constitutive matrix \( (C^l) \) a sum must be made:

\[
C^l = \sum_{i=1}^{n} w_i C_i = w_1 C_1 + w_2 C_2 + \cdots + w_n C_n, \quad 0 < w_i < 1
\]  

(1)

In equation (1), one can find the various fibre orientation constitutive matrixes \( (C_1, C_2, etc.) \) and the respective parameters \( w_i \) (weight functions). These are functions of the various design variables \( x_i \) and measure the relative contribution of each fibre orientation to the overall layer constitutive matrix \( (C^l) \). Also like the topology design variables, they take values from 0 to 1 because each fibre orientation constitutive matrix cannot contribute more than its own value to the global layer matrix and because negative values have no physical meaning.

In the case of this thesis, since the only parameter being optimized is the fibre orientation, the parameter \( n \) which can be seen in equation (1) represents the number of possible angles that the fibres can be laid in.

To guarantee the best performance possible, the optimizer should be able to drive the design variables, and therefore the weights functions, to the values of 0 and 1, in an efficient way, so that only one orientation is set per layer. This requires a good choice of weight functions \( w_i \), to ensure that a well-defined fibre orientation is obtained for each layer.

In [6], five possible parameterization schemes of the functional dependence of \( w_i \) are presented and their virtues and faults outlined. Based on this analysis only the 4th and 5th schemes will be described, as these are shown to be the best choices.
2.2.1. DMO Scheme 4 (DMO4)

As mentioned, the weight functions are used to in equation (1) to obtain the constitutive matrix as a sum of the contribution of the various fibre orientations. As the optimization proceed, the goal is to only have one of the weights $w_i$ equal to 1, the remaining having null value, for each layer, thus indubitably identifying the optimal fibre orientation in each ply. These weight functions are effectively auxiliary variables that relate the (primary) design variables in each layer $x_i^l$ with the intent of amplifying the variable’s drift to the extreme values of 0 and 1. This is achieved through properly designed equations such as the ones described in [6]. DMO4 is one of the most promising parameterization schemes and is described by:

$$C_i = \sum_{i=1}^{n} \left( (x_i^l)^p \prod_{j=1, j \neq i}^{n} \left[ 1 - (x_j^l)^p \right] \right) w_i$$

(2)

DMO4 is very efficient in driving the design variables to their limit values. This is achieved by the inclusion of the $(1 - x_j^l)$ term, so that an increase of $x_j^l$ leads to the decrease of all other weights.

The parameter $p$ is a penalty value, which is applied to help the weight functions drift to the limit values. As a rule of thumb, the larger the penalty value, the more the optimizer will avoid intermediate values.

The penalization effect on the design variables for scheme 4 can be seen in Figure 2-2, which shows the sum of the variables with different penalty values.

![Figure 2-2](image)

*Figure 2-2 – Image taken from [6] which represents the sum of two variables $w_1$ and $w_2$, for two materials, computed with DMO scheme 4.*

By observation of Figure 2-2, it becomes clear that the higher penalty value leads to larger areas in the corners of the graph, where the variables have value of either 1 or 0, corroborating the previous conclusions.
Scheme 4 does not assure that the sum of weights is unitary, which depending on the type of analysis, can have big impacts on the results obtained.

The main disadvantage of this method is that, because there is no guarantee of a unitary value for the sum of weights, the initial stiffness values of the layers can be unrealistically low. This does not cause any difficulties in compliance optimization problems but can be disastrous in cases that rely on realistic values of the stiffness and mass matrices, on every iteration, such as vibration and instability problems. The reason behind this is that low stiffness regions can lead to big variations of the \((k/m)\) relation, resulting in numerical problems.

For this reason alone this scheme should not be selected, as the 5th one proves more reliable. This parameterization scheme is outlined on the next subchapter.

2.2.2. DMO Scheme 5 (DMO5)

\[
C^i = \sum_{i=1}^{n} \frac{w_i}{\sum_{k=1}^{n} w_k} C_i \quad \text{where} \quad w_i = (x_i^1)^p \prod_{i=1, j \neq i}^{n} \left[1 - (x_j^1)^p\right]
\]  

Equation (3) results from a simple alteration of equation (2). By dividing the value of the weights for their sum one can guarantee that their sum shall always be unitary.

This allows for a fast convergence to near optimum points, but it also makes complete convergence much more difficult to achieve. This is due to the penalization being affected by the division, taking inferior values which cannot be countered by a simple increase of its value. Since the penalty is inferior, the number of favourable solutions increases, which leads to a higher number of iterations and overall computational time.

The idea of an incomplete convergence can seem very serious, but in reality it poses less of a problem than one would anticipate as the final values of the design variables can still clearly indicate the best solution. Variables that have high values, such as 0.9 or 0.8 point to the fact that these are indeed better solutions than those with low values such as 0.2.

This was the conclusion established in [6] and it is one that is corroborated by the results obtained when testing the application developed for this work.
2.3. Finite Element Analysis (FEA)

Finite element analysis was developed in 1943 by R. Courant and made use of the Ritz method to obtain approximate solutions to vibration systems. The concept was broadened in 1956 by Turner, Clough, Martin, and Topp [13].

Since then this technique has seen many developments which, coupled with the ever increasing computational power available, have resulted in several general purpose commercial finite element analysis programs such as Ansys, which are capable of yielding good results for highly complex systems.

In finite element analysis where one of the dimensions is much smaller than the other two, it is convenient to use shell elements [14]. This is a way of avoiding the enormous number of solid elements that would have to be used, to execute a comparable analysis. Shell elements are however susceptible to a number of problems such as locking [15]. Locking occurs when elements or meshes deform less than expected and is a prevalent problem in shell elements in both linear and non-linear analysis [16].

Locking can nevertheless be reduced as much as possible through careful choice of the element as well as its underlying equations. Nowadays, the most popular and simple finite element classes combine an equivalent single layer (ESL) laminate description, with finite elements based on the first order shear deformation theory (FSDT).

To reduce the computational weight of the analysis, linear elements are normally used. Since lower order elements are more susceptible to locking issues, an assumed natural strain technique (ANS) as described in [17] is employed and the elements originated by it are very robust.

The FEA program utilised on this work, Ansys, makes use of these techniques as do most commercially available programs.
3. Optimization

This section is devoted to the methods used to solve the optimization problem. It will also be used to explain the objective function choice and problem formulation as well as the sensitivity analysis.

3.1. Problem Formulation

As already stated, the main purpose of this work is to develop a computational methodology to maximise the fundamental frequency, namely in the case of a laminate flat plate, but easily extendable to other laminate type of structures. Since optimization problems should be about minimisation, and we’re trying to maximise, the objective function shall be adapted to convert a maximisation into a minimisation. This is achieved quite simply, by minimising the negative value of the fundamental frequency, which does not increase the complexity of the problem and therefore keeps computational cost as low as possible.

As we’re only concerned with fibre orientation, the plate’s mass will always be constant, rendering any mass constraints completely unnecessary. In fact, one finds that the only constraints required for this application are the limit values of the design variables.

Mathematically this is represented as:

\[
\begin{align*}
\text{Objective : } \max_x \lambda_l &= -\min_x -\lambda_l \\
\text{subject to : } \underline{x} \leq x \leq \overline{x}
\end{align*}
\] (4)

To solve the system, a sensitivity analysis shall have to be performed.

3.2. Sensitivity Analysis

In order to perform a sensitivity analysis, the eigenvalue problem shall formulated.

3.2.1. Eigenvalue Problem

For a multiple degrees of freedom system the equations of motion should be written in compact matrix form that follows:

\[
[K][x] + [C][\ddot{x}] + [M][\dot{x}] = \{F\}
\] (5)

Considering a free and undamped vibrating system, both the damping and force terms are null.
To solve the equation we assume a displacement (homogeneous solution) in the form of:

$$\{x\} = \{X\} e^{i\omega t}$$  \hspace{1cm} (7)

Which after substitution in (6) leads to:

$$\{[[K] - \omega^2[M]]\{X\} e^{i\omega t} \} = 0$$  \hspace{1cm} (8)

Since $e^{i\omega t}$ is a non-null term the final condition requires:

$$\{[[K] - \omega^2[M]]\{X\} \} = 0$$  \hspace{1cm} (9)

Possibility of non-zero solutions requires that the linear system matrix is singular \(i.e.\) we have an eigenvalue problem where the eigenvalue is identified with the square of the natural frequency $\omega^2$ and the eigenvector with the vibration mode $\{X\}$. These can be rewritten as:

$$\{[[K] - \lambda_i[M]]\phi \} = 0$$  \hspace{1cm} (10)

3.2.2. Lagrangian and Sensitivity Analysis

Assuming implicitly the “equilibrium” constraint (10), the Lagrangian associated with the optimization problem (4) can be given as:

$$L = -\lambda_i + \sum_i \left[U_{1i}(x - x_i) + U_{2i}(x_i - \bar{x})\right], \quad i = 1, \ldots, n$$  \hspace{1cm} (11)

With $U_{1i}$ and $U_{2i}$ identifying the Lagrange multipliers associated with each bound constraint and $n$, the number of design variables \(i.e.\) possible fibre orientations times the number of plies ($n = i \times l$).

Assuming the Lagrangian (11), the KKT optimality (necessary) conditions require stationarity with regard to the design variables. Thus from:

$$\frac{\partial L}{\partial x_i} = -\frac{\partial \lambda_i}{\partial x_i} - U_{1i} + U_{2i}$$  \hspace{1cm} (12)

We get the necessary condition:

$$-\frac{\partial \lambda_i}{\partial x_i} - U_{1i} + U_{2i} = 0$$  \hspace{1cm} (13)
Since in general the optimizers deal with the side constraints internally, they can be left out of the equation yielding the necessary condition:

\[
\begin{align*}
\frac{\partial L}{\partial x_i} &= -\frac{\partial \lambda_i}{\partial x_i} = 0 \text{ if } x_i \neq \bar{x} \text{ or } \bar{x} \\
\frac{\partial L}{\partial x_i} &= \frac{\partial \lambda_i}{\partial x_i} \geq 0 \text{ if } x_i = \bar{x} \quad \quad i = 1, \ldots, n \\
\frac{\partial L}{\partial x_i} &= -\frac{\partial \lambda_i}{\partial x_i} \leq 0 \text{ if } x_i = \bar{x}
\end{align*}
\] (14)

It becomes clear that the derivative of the natural frequency $\frac{\partial \lambda_i}{\partial x_i}$ must be defined in order to explicitly state the necessary condition and solve the optimization problem. To do so, we differentiate equation (10) (thus imposing the implicit constraint realization through the optimisation process):

\[
\left[ \frac{\partial [K]}{\partial x_i} - \left( \lambda_i \frac{\partial [M]}{\partial x_i} + \frac{\partial \lambda_i}{\partial x_i} [M] \right) \right] \phi + ([K] - \lambda_i [M]) \frac{\partial \phi}{\partial x_i} = 0 \iff (15)
\]

\[
\iff \left[ \frac{\partial [K]}{\partial x_i} - \left( \lambda_i \frac{\partial [M]}{\partial x_i} + \frac{\partial \lambda_i}{\partial x_i} [M] \right) \right] \phi = -([K] - \lambda_i [M]) \frac{\partial \phi}{\partial x_i} \iff (16)
\]

By multiplying the previous equation by $\phi^T$ and recalling equation (10), the expression can be further simplified as follows:

\[
\phi^T \left[ \frac{\partial [K]}{\partial x_i} - \left( \lambda_i \frac{\partial [M]}{\partial x_i} + \frac{\partial \lambda_i}{\partial x_i} [M] \right) \right] \phi = -\phi^T([K] - \lambda_i [M]) \frac{\partial \phi}{\partial x_i} \iff (17)
\]

\[
\iff -\phi^T \frac{\partial \lambda_i}{\partial x_i} [M] \phi = \phi^T \left( -\frac{\partial [K]}{\partial x_i} + \lambda_i \frac{\partial [M]}{\partial x_i} \right) \phi \iff (18)
\]

\[
\iff \frac{\partial \lambda_i}{\partial x_i} = \frac{\phi^T \left( \frac{\partial [K]}{\partial x_i} - \lambda_i \frac{\partial [M]}{\partial x_i} \right) \phi}{\phi^T [M] \phi} \iff (19)
\]

By ensuring that the eigenvectors are mass normalized, equation (19) can be further simplified, as the lower terms on the division take unitary value.

\[
C_i = \phi^T [M] \phi = 1 \iff (20)
\]
\[
\frac{\partial \lambda_i}{\partial x_i} = \phi^T \left( \frac{\partial [K]}{\partial x_i} - \lambda_i \frac{\partial [M]}{\partial x_i} \right) \phi
\]  

(21)

With the sensitivity deduced all the mathematical information required for the optimization process has been laid out.

### 3.3. Brief description of the optimizer used

To solve the problem a third party optimizer was used, in this case, a versatile program written by Prof. Herskovits called FAIPA [18]. Other possible choices for the optimizer include the Method of Moving Asymptotes (MMA) [19] and Matlab’s fmincon(.) function.

FAIPA and FDIPA (Feasible Direction Interior Point Algorithm) are interior point algorithms for the minimization of a nonlinear function with equality and inequality constraints. These consist on fixed point iterations to solve the KKT (Karush-Kuhn-Tucker) conditions [18].

In FDIPA, a descent direction is defined at each iteration by solving a linear system. This system is then perturbed to deflect the descent direction and obtain the feasible descent direction. This method ensures global convergence to the KKT points but can also display slow convergence in problems with highly nonlinear constraints. FAIPA attempts to avoid this problem by defining a curvilinear search inside the feasible region [18].

Both these algorithms can be based on Newton, Quasi-Newton or first order methods. Since the problem approached in this work is an unconstrained one, both algorithms function in the same way and will therefore yield equal results. However, by using FAIPA, one can be certain that if a more complex (constrained) case is studied in the future, convergence shall be yielded efficiently.

The optimizer also features finite difference calculation, a powerful tool for cross-checking the derivatives and making sure they have been well calculated.

In this work, the quasi-Newton version of the algorithm is employed, along with a line search based on the Wolfe conditions.

When solving an optimization problem one must find the best balance possible between convergence speed and computational cost. In many cases, a method that has intermediary characteristics will be the best option. While the steepest descent method is a great example of a low-cost but slow convergent method, the Newton method has the opposite behaviour, guaranteeing excellent convergence characteristics but high computational cost, chiefly because it requires the calculation of the Hessian matrix.

Ideally then the method used should combine the best characteristics of both methods. This is achieved through the application of the quasi-Newton method.
3.3.1. Quasi-Newton method

The quasi-Newton method (in reality a family of methods) is based on Newton’s method and attempts to locate a function’s global minimum. It differs from the Newton’s method in that it does not require the hessian matrix information.

The method revolves around a search direction, calculated through the use of an approximate hessian matrix \( B_k \). This Hessian approximation is iteratively constructed using gradient and design variables information gathered along the iterative process. In order to obtain \( B_k \), an update scheme such as Broyden-Fletcher-Goldfarb-Shanno (BFGS) or Davidon-Fletcher-Powell (DFP) will need to be employed [20], [21], [22]. This approximate hessian matrix \( B_k \) should be:

- Non-singular, so that the search direction is well defined;
- Symmetric, like all Hessian matrices are;
- Positive definite so that the search direction is a descent one.

Once the search direction has been computed, the Quasi-Newton method, like the Newton and steepest descent methods, makes use of a line search to obtain the minimum point on the search direction selected.

![Figure 3-1 – Image taken from [23]: Steepest Descent Steps](image)

In order to have a better understanding of the method one should look at its algorithm:

1. Choose a non-singular \( B_k \) and a start point \( x_0 \in \mathbb{R}^n \). A common choice for \( B_k \) is the identity matrix;
2. Check if the absolute value of the difference between consecutive values of the function is inferior to the tolerance value i.e. \( \| \nabla f(x_k) \| \leq \epsilon \). If so, then \( x_k \) is the approximate value of the minimum. Otherwise continue;
3. Compute the search direction \( d_k = -B_k^{-1}\nabla f(x_k) \);
4. Perform a line search minimization \( \min_{\alpha > 0} \theta(\alpha) \) with \( \theta(\alpha) = f(x_k + \alpha d_k) \). Then define \( x_{k+1} = x_k + \alpha_k d_k \);
5. Calculate the new approximate Hessian by the method chosen (BFGS in this work);
6. Return to point 2.
3.3.2. Line Search – Armijo, Wolfe and Goldstein Conditions

As mentioned earlier, each line search iteration takes the search (descent) direction computed (in step 3) and decides how far to move along it.

\[ x_{k+1} = x_k + \alpha_k d_k \quad (22) \]

When choosing the line search length \( \alpha_k \), we are confronted with the dilemma: What should the ideal step length be, to get the largest function’s value reduction possible?

Finding this can be very computationally expensive and time consuming as the number of evaluations of the function’s value increases.

The ideal choice of \( \alpha \) would be the global minimizer of the function \( \varphi(\alpha) = f(x_k + \alpha_k d_k) \) (see Figure 3-2). Using this methodology is nevertheless, too computationally expensive to be a good way to solve the dilemma.

A more practical approach to this problem is undertaken with inexact line search methods. These aim at achieving an adequate reduction of the function’s value at reduced computational cost.

For an inexact line search to be efficient one must select an adequate stopping criteria. A very common line search condition imposes that \( \alpha_k \) should lead to an at least sufficient decrease of the objective function \( f \) according to the following inequality [23]:

\[ f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T d_k \quad where \ c_1 \in (0,1) \quad (23) \]

Essentially, the decrease in the objective function’s value should at least be proportional to both the step length and the directional derivative \( \nabla f_k^T d_k \).

This is called the Armijo condition and is one of the conditions that can be selected for optimizing the problems approached in this work.
We can denote inequation (23) right hand side, which is a linear equation, as \( l(\alpha) \). This function has a negative slope but for small values of \( \alpha \) and because \( c_1 \in (0,1) \), it always stays above the graph of \( \phi(\alpha) \).

![Figure 3-3 – Image taken from [23]: Sufficient decrease or Armijo condition](image)

This condition by itself is not sufficient to guarantee a good rate of progress, as any small value of \( \alpha \) could satisfy it. In order to ensure a decent rate, a second condition called curvature condition [23] must be introduced which requires \( \alpha_k \) to be:

\[
\nabla f(x_k + \alpha_k d_k)^T d_k \geq c_2 \nabla f_k^T d_k \quad \text{where } c_2 \in (c_1, 1)
\]  

(24)

The main objective of this condition is to ensure that there is a sufficient slope of \( \phi(\alpha) \) to guarantee that, by continuing the iteration process, a reasonable reduction of the value of the objective function will be obtained. By combining equations (23) and (24) one obtains the **Wolfe conditions**.

Another possible choice is the **Goldstein Condition**. Like the **Wolfe condition** it too attempts to ensure sufficient decrease of the objective function while avoiding step sizes that are too small. This is expressed with the following pair of inequalities [23]:

\[
f(x_k) + (1 - c)\alpha_k \nabla f_k^T d_k \leq f(x_k + \alpha_k d_k) \quad \text{(25)}
\]

\[
f(x_k + \alpha_k d_k) \leq f(x_k) + c\alpha_k \nabla f_k^T d_k \quad \text{(26)}
\]

Where \( c \in (0, \frac{1}{2}) \).

The first inequality controls the step length while the second one ensures a sufficient decrease condition.
4. Composite Materials

This chapter is dedicated to composite materials, their characteristics, production methods and design guidelines.

4.1. Material properties

Composite materials are obtained by combining two or more materials with different properties. Due to this combination, a new material is obtained, with significantly different properties (ideally combining the best of its individual constituents). This means that composites can have excellent mechanical properties, be stronger and lighter than conventional materials, sometimes at a reduced cost. More information on composite materials and its properties can be found in [24] and [25].

These two constituent materials are classified as the resin (see Figure 4-1) and the substrate. The substrate can have many shapes but in fibre reinforced composite materials, it takes the form of short or long fibres.


The resin’s function is to spread the loads applied on the composite between the individual fibres, as these have high strength properties and set the component’s structural resistance. The resin also serves to keep the fibres in their desired place and to impart some ductility and tenacity to the overall properties of the material, reducing the fibre’s brittleness. Resin can be polymeric, metallic or ceramic.

Metallic resins have good stiffness properties and high fusion point but are very heavy. If on the other hand it is made of polymeric material, then the melting point is much lower, as are the mechanical properties. The upshot is the reduced overall weight of the structure. Finally there are the ceramic resins which are extremely heat resistant but also very brittle.
The fibres can be either long or short depending on the intended material characteristics. In this case, we shall only look into continuous fibres since those are the ones that allow an orientated layered structure required for this particular optimization analysis. These fibres are made from high strength material and impart excellent properties of stiffness to the component but, because of this they are also usually brittle. This brittleness should be counteracted by the matrix’s more ductile properties.

Due to the great combination of properties that composite materials display and to the continuously reducing production costs, their usage has been steadily increasing in the last years. The great engine of this increase is of course, the aerospace industry, where metals are becoming more and more obsolete in most structural components. A great example of this is the Boeing 787 Dreamliner in which 50% of the overall weight is due to composite materials.

By comparison, on the Boeing 777, only 12% of the total weight was due to these sorts of materials.

Layered fibre reinforced composites can have its fibres oriented in any direction between \(-90^\circ\) and \(90^\circ\) but usually only a few standard angles are used such as \((0^\circ, \pm 45^\circ, 90^\circ)\) [10]. This standardization can be explained by the fact that an unconventional fibre direction will necessitate additional laboratory tests and analysis. Choice of the standard orientations also allows reduced personnel training and manufacturing costs. Even though these directions may not necessarily represent the best possible solution, they provide the means to obtain good or excellent results, while greatly reducing production costs.

The fact that the fibres are oriented according to a certain direction means that the overall component will have anisotropic behaviour, where its mechanical properties are directionally dependent. This is the reason why careful choice of fibre orientation is so important, as the maximum resistance of each layer is attained only when loads act on the direction of the fibres.
4.2. Manufacturing Technologies

Nowadays there are several different production techniques, for the manufacture of continuous fibre reinforced composite materials [24]. The main ones are:

- Hand lay-up;
- Vacuum bagging;
- Filament-winding;
- Pultrusion;
- Resin transfer moulding (RTM).

4.2.1. Wet Lay-up/Hand Lay-up

Wet lay-up is the most basic composite material production technique. It starts with the application of a gel coat and release film to the internal surface of the mould to facilitate the posterior component removal. A layer of woven-fibre is then carefully placed to ensure desired fibre orientation. Resin is then poured onto the fibres and spread with a roller to ensure even distribution. The process is then repeated until the desired number of layers has been set. It's finished by curing it either with heat, ultraviolet light or at atmospheric conditions.

This process is ideal for small scale parts and production as it is a slow process. Components produced in this way can have high fibre percentage. However, the overall mechanical properties and quality are very dependent on the skill of the laminator.

![Wet lay-up/Hand lay-up](image)

*Figure 4-3 – Wet Lay-up production technique. Image from [26].*

4.2.2. Vacuum bagging

Vacuum bagging is a variation of the wet lay-up process and is used to improve the overall quality of the components produced. Most of the process methodology is the same as wet lay-up. However, after all the layers are stacked together, a thin bagging film is placed around the
component and then sealed to the tool’s surface. The air is subsequently removed until vacuum is achieved, causing the component to suffer a pressure up to 1 atmosphere.

The advantage of this method when compared with the hand lay-up technique is that it allows a more even distribution of resin and reduces the number of voids present on the final composite material resulting in higher product quality. This added quality comes at the price of extra production costs.

**Vacuum bagging**

![Diagram of vacuum bagging process](image)

*Figure 4-4 – Vacuum bagging manufacturing technique. Image from [26].*

### 4.2.3. Filament-winding

Filament-winding is the ideal technique for creating cylindrical or other axisymmetric components. In this process the fibre is fed from a spool to a rotating mandrel after passing through a resin impregnation bath. By means of a special carriage the fibre is lead onto the mandrel allowing for a variety of angles to be achieved. After sufficient material has been applied the mandrel is removed for curing, usually in an autoclave.

![Diagram of filament winding process](image)

*Figure 4-5 – Filament winding process. Image from [26].*

The finished component is a hollow cylinder, ideal for pipes, pressure vessels or tanks as it has excellent hoop strength.
4.2.4. Compression Moulding

In compression moulding a composite material charge is placed between a two-part mould. This charge is subsequently heated and compressed taking the negative image of the mould and curing under pressure. This process is ideal for high volume manufacture not only due to the quick production speed but also because of high tooling costs. Components obtained by this process have high quality and excellent surface finish.

![Compression Moulding](image)

_Figure 4-6 – Compression moulding method. Image from [26]._

4.2.5. Pultrusion

In the pultrusion method, fibres or a prepreg unidirectional tape, are continuously pulled through a resin matrix bath onto a performer which will give it the desired cross-sectional shape. A set of pre-heated dies are then used to finalize the cross-sectional form, to remove any excess resin that may exist and to cure the composite material in preparation for the final step of cutting it to the desired length through means of a cut-off saw.

![Pultrusion](image)

_Figure 4-7 – Pultrusion production technique. Image from [26]._

This process is ideal for large production volume of components as it works continuously. It can produce a variety of profiles such as flat plates, tubes, cross-section beams and sections and parts produced by this technique display excellent material properties due to high fibre percentage and good resin dispersion.
4.2.6. Resin Transfer Moulding (RTM)

In this process, reinforcement material or fabrics, are put into a set of mould halves which are subsequently closed. Resin and catalyst are then fed into the mould, either gravity fed or pumped, until they fill the whole cavity and permeate the fibres. The component then cures whilst in the closed mould, before being removed.

![Resin transfer moulding diagram](image)

*Figure 4-8 – Resin transfer moulding. Image from [26].*

This process is somewhat related to the lay-up methods but displays a greater degree of automation which allows for faster production times. Despite this improvement it is still a slow manufacture process, ideal for the production of large components which can have complex shapes. For these reasons it presents itself as the ideal choice for the manufacture of boat hulls, automobile components among other large components.
5. Computational model

This chapter will discuss the computational analysis and design choices involving the finite elements and its underlying plate theory.

5.1. Plate Theory

Laminate composite materials are usually treated as plate or shell elements and as such, need to be approached through a plate theory, of which there are many. This subchapter is dedicated to a brief description of some of these methods.

Structural plate theories can be grouped in 3 distinct classes [27] as:

1. Equivalent single-layer theories - ESL (2-D)
   - Classical Laminated Plate Theory (CLPT);
   - Shear Deformation Laminated Plate Theories;
2. Three-Dimensional Elasticity Theory (3-D)
   - Traditional three-dimension (3-D) elasticity formulations;
   - Layer wise theories;
3. Multiple model methods (2-D and 3-D).

In ESL theories, a three-dimensional heterogeneous plate is treated as a single statically equivalent layer, i.e. the three-dimensional problem is converted into a bi-dimensional one. This is done through a series of assumptions that are related with the kinematics of deformation or with the stress state along the laminate's thickness.

Three-dimensional theories approach the problem in three dimensions and so each layer is treated as a solid element.

For computational lightness and simplicity ESL type theories are very attractive as the bi-dimensionality greatly reduces the number of nodes required on the model. As such only these theories shall be outlined. The keen reader may find out more about these and other plate theories in [27].

5.1.1. ESL – Classical Laminate Plate Theory (CLPT)

CLPT is the most basic of the ESL theories and consists of an extension of Kirchhoff’s classical theory to laminate composite plates. As such it is based on a series of rules called Kirchhoff’s hypotheses:
- Straight lines perpendicular to the mid plane before deformation, must still be straight after it;
- Transverse normals do not elongate;
- Transverse normals rotate to remain perpendicular to the mid surface.

In CLPT transverse shear and normal stress effects are ignored and deformation is due to bending and in-plane stretching. A final simplification is made by assuming the material is subject only to plane-stress.

Although a deduction shall not be provided in this work one can be found in [27], where one finds that the theory is based on the following field of displacements:

\[
\begin{align*}
    u(x, y, z, t) &= u_0(x, y, t) - z \frac{\partial w_0}{\partial x} \\
    v(x, y, z, t) &= v_0(x, y, t) - z \frac{\partial w_0}{\partial y} \\
    w(x, y, z, t) &= w_0(x, y, t)
\end{align*}
\]  

(27)

Where \((u_0, v_0, w_0)\) are the displacements and \((x, y, z)\) the coordinate directions.

![Figure 5.1 - Undeformed and deformed geometry under the Classical Laminate Theory. Picture from [27].](image)

5.1.2. First-Order Shear Deformation Theory (FSDT)

FSDT is an extension of the classical theory in which the Kirchhoff hypothesis has been relaxed by removing the third condition. This means that the transverse normals no longer need to remain perpendicular to the mid surface after deformation which amounts to including transverse shear strain in its kinematics assumptions.
Under the remaining assumptions and hypothesis as CLPT the displacement field is now:

\[
\begin{align*}
    u(x, y, z, t) &= u_0(x, y, t) + z\phi_x(x, y, t) \\
    v(x, y, z, t) &= v_0(x, y, t) + z\phi_y(x, y, t) \\
    w(x, y, z, t) &= w_0(x, y, t)
\end{align*}
\] (28)

Where \(\phi_x\) and \(\phi_y\) are the rotations of a transverse normal along the \(y\) and \(x\) directions respectively.

The main difference between the FSDT and CLPT theories is that the transverse shear strain is not null and is constant along the thickness direction, resulting in a more accurate modelling of the plate’s behaviour. The flip side of this is the higher computational cost of the model.

Other ESL theories exist and are called higher order theories. These make use of higher order polynomials, in the expansion of the displacement components along the plate’s thickness. The added complexity results in a better accuracy of the results but despite this, Ansys’ elements use FSDT to model its shell elements as that model allows for the best ratio between accuracy and computational calculation speed.

5.2. Finite Elements

In order to be able to produce the results needed for the optimization process, a FEA shall be executed with Ansys. This FEA program contains various different finite elements which have been grouped together in different “types” such as solid, beam and shell, among others.

Since this work’s focus is on flat plate optimization, where one dimension is much smaller than the other two, shell type elements are the most obvious choice. Usage of shell elements
reduces the computation cost of the analysis as no elements are needed along the thickness direction, something that would not hold true if for example, solid elements were to be used.

*Ansyl* features several different types of shell elements [28] as listed below:

- 3-D Shell elements – With 4 or 8 nodes;
- Axisymmetric shell elements – With 2 or 3 nodes;
- Shear panel elements – Shell28;
- Membrane elements – Shell41;
- Axisymmetric-harmonic elements – Shell61.

Axisymmetric and axisymmetric-harmonic shell elements are used for shapes with axisymmetry, such as tubes or reservoirs. They are therefore not ideal for a case where the geometry is flat and can be eliminated from the list.

![Figure 5-3 – Axisymmetric geometry modelled with Ansys. Image from http://www.mece.ualberta.ca/tutorials/ansys/IT/Axisymmetric/Axisymmetric.html](http://www.mece.ualberta.ca/tutorials/ansys/IT/Axisymmetric/Axisymmetric.html)

Shear panel elements are designed to carry shear load in a frame structure and only have 3 degrees of freedom at each node: translation in the nodal x, y and z directions. Since no shear loads will be present on the vibration analysis these are not a good choice.

Membrane shell elements are meant for cases in which the element bending is of secondary importance since the element has no bending stiffness. These elements only have 3 degrees of freedom at each node: translations in the nodal x, y and z directions. As in vibration analysis bending is of importance these elements are not an adequate choice.

We are left with the 3-D type shell elements which have 6 degrees of freedom per node (3 translations and rotations) and are adequate for modelling thin to moderately thin plate structures. Of these there are two possible choices: 4 node elements (*Ansyl* element 181) or 8 node elements (*Ansyl* element 281).

The 4 node element’s main advantage is the reduced computational cost due to the existence of much lesser number of nodes in the system. They may however generate problems such as shear locking effects.
Shear locking effects occur for several reasons. One of the main culprits is the fact that the 4 node element cannot accurately model the curvature of the element under deformation. This leads to the presence of additional shear stress, to smaller displacements and therefore an unrealistically high element stiffness.

![Figure 5-4 – Shear locking of a linear element. On the left is the realistic deformation and on the right the one obtained with these elements. Image found in http://www.sintef.no/project/eVITAmeeeting/2012/KMM%20Geilo%202012%20Lecture%202010.pdf](http://www.sintef.no/project/eVITAmeeeting/2012/KMM%20Geilo%202012%20Lecture%202010.pdf)

Elements 181 and 281 available in Ansys make usage of the FSDT that was briefly outlined in chapter 5.1.2.

The 4 node element 181 also incorporates an assumed shear strain formulation of Bathe-Dvorkin [17], to alleviate the locking effects. This allows for better results when using the 4 node element while still guaranteeing low computational cost.

Due to representing the best relation between accuracy and computational lightness, the analysis shall be executed using the linear shell element 181.

For more information on Ansys programming and its elements and background theories, refer to [28], [29], [30], [31].

### 5.3. Penalty Selection

The DMO method relies on penalization in order to drive the design variables to the limit values of 1 and 0. This penalization is applied thanks to the term $p$ that can be seen in equation (3) and an adequate choice of its value is essential to guarantee the most efficient optimization possible.

Theoretically, this penalty term can have any value possible and the larger it is the more it discourages “grey” results i.e. design variables with intermediary values. Certain guidelines should, however, be respected whenever possible to ensure adequate results.
• The penalty should have low values for the first few iterations, as high values shall encourage the program to start pushing the variables in big jumps, which can lead to unstable behaviour and bad results;
• Penalization should afterwards be slowly increased to higher values to discourage intermediate values;
• A maximum value should be set, to avoid exceedingly large values appearing when the optimization process goes on for a large number of iterations. Since $p$ affects not only the weight values but also directly the derivatives, a large penalty value can originate large derivative values which can lead to weak optimizer performance (the maximum value of the derivatives should whenever possible have a value as close to 1 as possible).

Following these guidelines, testing was done applying the following penalization scheme:

\[
\begin{align*}
p &= 1 \text{ when } \text{iteration} \in [1,2] \\
p &= 2 \text{ when } \text{iteration} \in [2,8] \\
p &= 4 \text{ when } \text{iteration} \in [8,16] \\
p &= 6 \text{ when } \text{iteration} \in [16, \infty]
\end{align*}
\]  

(29)

In some cases, scheme (30) did not yield the expected performance due to phenomena such as mode switching or to DMOS’s shortcomings. To improve the optimization performance in these troublesome examples, different penalization schemes can lead to better results. Test case 4 in chapter 6 is an example of this.

### 5.4. Mode of Vibration Superposition and Switching

Before proceeding to the program’s algorithm and to the analysis of the results, special attention must be given to the particular phenomenon of vibration called superposition of modes of vibration and mode switching.

#### 5.4.1. Mode Superposition

Mode superposition occurs when there are two or more distinct modes of vibration (eigenvectors) for a single natural frequency (eigenvalue), and is a prevalent phenomenon in plate, shell or membrane models. A good example of this effect is a simply supported square plate in which the natural frequencies and mode shapes are respectively obtained from [32]:

\[
\omega_{mn} = \left(\frac{\pi}{a}\right)^2 (m^2 + n^2) \frac{D}{\rho h}
\]  

(30)
Where \( m \) and \( n \) can have any value from positive integers \((m, n = 1,2,3,\ldots)\).

\[
U_{3mn} = A_{mn} \frac{m \pi x}{a} \sin(n \pi y)
\]  

(31)

From equations (30) and (31), it becomes clear that different combinations of \( m \) and \( n \) that yield the same results with \((m^2 + n^2)\), will have the same natural frequency, but different modes of vibration. This is therefore a typical case of mode superposition.

This phenomenon can have a very big impact on the optimization’s performance as it makes it difficult or even impossible for the used optimizer to work.

As stated in chapter 3.3, the optimizer works by using the quasi-Newton method. In the quasi-Newton method, the search direction is computed using the objective function’s derivatives, on a given point of the domain. These derivatives are calculated through equation (21), which requires input from the eigenvectors to work.

If mode superposition happens, the objective function (fundamental frequency) is only directionally differentiable (see [33]). This means that optimization methods based on gradient information are not the ideal choice.

### 5.4.2. Mode Switching

Another important vibration phenomenon is called mode switching and it occurs when two modes of vibration switch positions. If two consecutive frequencies, during the optimization process, have similar values, mode switching is very likely to occur; as the value variation of the frequencies is irregular (the fundamental frequency’s value increases faster than the second natural frequency in several test cases). If this happens than the second frequency can take a smaller value than the first one, leading to the second mode of vibration being perceived by the optimizer as the one corresponding to the first natural frequency of vibration.

When this phenomenon happens, the eigenvector used to calculate the derivative will be radically different which can lead to a change in the search direction. On the next iteration the eigenvalues can then cross once more leading to a derivative that is very similar to the one obtained 2 iterations ago. The process can then repeat itself and the optimization cycle enters an endless loop.

To solve such a problem, a whole different approach must be undertaken when devising the sensitivity analysis and optimization methods. Such an approach will not be shown or used in this work but it requires a different sensitivity analysis such as the one in [33] as well as a slightly different optimization algorithm.
5.5. Algorithm

This chapter will be dedicated to explaining the program’s functionality. Special attention shall be given to the various design choices made during the code writing phase and an algorithm scheme shall also be provided for easier understanding.

5.5.1. Program’s intended functionalities

In order to better understand the design choices and the way the program works, a brief outline of the initial intended program capabilities and overall functionality is listed:

1. The program should be able to select an appropriate mesh size, to ensure that the finite element results converge;
2. The program should be able to use a 3rd party FEA program, such as Ansys, directly from Matlab to execute the problems’ structural analysis;
3. Ansys should be called in batch mode directly from Matlab, read a log code with the analysis instructions and output the results in text files;
4. Ansys’ input log files should be written by Matlab, so that any changes made by the user to the initial conditions are also automatically done to the log files;
5. Matlab should be able to read the input files created by Ansys and store them in appropriate variables in the workspace;
6. The optimization process shall be done using a 3rd party optimizer (FAIPA) to ensure the most robust and efficient analysis process.

5.5.2. Overview of the program’s structure

In order to aid in the understanding of the program’s inner workings and the reasons behind them, a simplified algorithm is presented in (Figure 5-5).
From the algorithm one can identify 3 main program’s sections:

- Pre-processing;
- Optimization (Consisting on the blocks in blue);
- Post-processing.

5.5.2.1. Pre-Processing

The pre-processing phase contains all the operations necessary before entering the optimization proper and makes usage of both Matlab and Ansys capabilities. Matlab starts by declaring and initializing the various variables required for the rest of the program, including the problem’s geometry and variables.

A mesh convergence analysis is then run using Ansys in batch mode, in an attempt to find the best compromise between obtaining accurate results and computation speed. In order to make sure the process is as robust as possible, lower and upper bound limits to the mesh size are imposed, so as to guarantee that the mesh is never either too fine or too coarse.

Once a suitable mesh has been devised, Matlab will once again call Ansys and proceed to mount the stiffness and mass matrices. The initial goal was to extract, from Ansys, all the structural information necessary for the differentiation of the problem at each iteration. To achieve this, Ansys must be given a 21 entry anisotropic stress-strain matrix \([D]\), that has to be previously determined by the user, in the proper coordinate system. Documentation as to its
calculation is however scarce and apparently incomplete and therefore an adequate transformation matrix could not be devised, making the whole idea impracticable.

The general steps required to better understand how to get an anisotropic stress strain matrix, using the notation in [27], are:

\[
\begin{bmatrix}
\sigma_1 \\
\sigma_2 \\
\sigma_3 \\
\sigma_4 \\
\sigma_5 \\
\sigma_6
\end{bmatrix}
= \begin{bmatrix}
C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\
C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\
C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\
C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\
C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\
C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66}
\end{bmatrix}
\begin{bmatrix}
\varepsilon_1 \\
\varepsilon_2 \\
\varepsilon_3 \\
\varepsilon_4 \\
\varepsilon_5 \\
\varepsilon_6
\end{bmatrix}
\]

(32)

With the matrix \([C]\) being the stress-strain matrix.

By assuming that the material is hyperelastic one can simplify this matrix, as this implies that the coefficients \(C_{ij}\) are symmetric. This leads to:

\[
\begin{bmatrix}
\sigma_1 \\
\sigma_2 \\
\sigma_3 \\
\sigma_4 \\
\sigma_5 \\
\sigma_6
\end{bmatrix}
= \begin{bmatrix}
C_{11} & C_{21} & C_{31} & C_{41} & C_{51} & C_{61} \\
C_{21} & C_{22} & C_{32} & C_{42} & C_{52} & C_{62} \\
C_{31} & C_{32} & C_{33} & C_{43} & C_{53} & C_{63} \\
C_{41} & C_{42} & C_{43} & C_{44} & C_{54} & C_{64} \\
C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\
C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66}
\end{bmatrix}
\begin{bmatrix}
\varepsilon_1 \\
\varepsilon_2 \\
\varepsilon_3 \\
\varepsilon_4 \\
\varepsilon_5 \\
\varepsilon_6
\end{bmatrix}
\]

(33)

In order to obtain the appropriate matrix \([D]\) that Ansys requires, a coordinate transformation step must be undertaken so that this new matrix \([D]\) relates the stress and strain in terms of the global coordinates \(x, y, z\), in the order used in Ansys, instead of the principal directions 1, 2, 3. As mentioned the correct coordinate transformation could not be devised but by using [27] as a reference, such a matrix is called \([\bar{C}]\) and can be obtained by doing:

\[
[\bar{C}] = [T][C][T]^T
\]

(34)

Where \([T]\) is:

\[
\begin{bmatrix}
\varepsilon_{xx} \\
\varepsilon_{yy} \\
\varepsilon_{zz} \\
\varepsilon_{yz} \\
\varepsilon_{xz} \\
\varepsilon_{xy}
\end{bmatrix}
= \begin{bmatrix}
cos^2\theta & \sin^2\theta & 0 & 0 & 0 & -\sin\theta\cos\theta \\
\sin^2\theta & \cos^2\theta & 0 & 0 & 0 & \sin\theta\cos\theta \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & \cos\theta & \sin\theta & 0 \\
0 & 0 & 0 & \sin\theta & \cos\theta & 0 \\
\sin 2\theta & -\sin 2\theta & 0 & 0 & 0 & \cos^2\theta - \sin^2\theta
\end{bmatrix}
\begin{bmatrix}
\varepsilon_1 \\
\varepsilon_2 \\
\varepsilon_3 \\
\varepsilon_4 \\
\varepsilon_5 \\
\varepsilon_6
\end{bmatrix}
\]

(35)

We can therefore rewrite (32) as:
Obtaining \([D]\) should be a simple case of doing:

\[
[D] = [\hat{T}] [C][\hat{T}]^T
\]  

(37)

Where \([\hat{T}]\) is the correct matrix of transformation used internally by Ansys.

To tackle this problem, the program’s algorithm was changed, to use Ansys as few times as possible. Instead of being used for all the structural calculations, Ansys shall only be used for obtaining the stiffness \([K]\) and mass \([M]\) matrixes for each orientation in each layer.

If one imagines an optimization case where a flat plate with two layers and four possible fibre orientations is optimized, then eight matrixes \([K]\) and eight matrixes \([M]\) are required to be able to solve the optimization problem. Since the distance between a layer and the reference plane has a big impact in its stiffness, a simple trick has to be used to obtain the accurate stiffness matrixes. While collecting the information relative to one particular layer, all remaining layers are filled with a material with almost null properties, to simulate the layer’s distance to the reference plane without having unintended stiffness and mass contributions from the remaining layers. This means that stiffness and mass information can be collected safely with little error.

```
Figure 5-6 – Simple optimization problem with 2 layers and the reference plane displayed.
```

To exemplify this procedure, the optimization case in Figure 5-6 shall be used. To solve this optimization problem, four \([K]\) and \([M]\) matrixes need to be collected, for each layer. To achieve this goal, when collecting the data for layer 1, void material is given to layer 2 and the opposite happens when the matrixes for layer 2 are being collected.

The matrix information is then written to text file in Harwell-Boeing (HB) format [34], which is a compressed matrix format, using the code in [35]. This information is then read into Matlab in sparse format and stored in an appropriate cell array for subsequent use. From then on, Ansys is no longer required.
Using the DMO formulation 5 which has already been detailed in equation (3) of this thesis, the program then calculates the various weight function’s initial values \( (x_j^l) \).

Once all this data has been collected the program is ready to start the optimization phase.

A small block scheme is displayed in Figure 5-7, which summarises what was stated earlier.

**Pre-Processor**

5.5.2.2. **Optimization**

Optimization starts with the calculation of \([K]\) and \([M]\), in sparse format, for the whole plate structure. This is achieved by multiplying the weight values \((x_j^l)\) with the respective orientation and layer matrixes \([K_j^l]\) and \([M_j^l]\) and then by summing the results (in a similar way to the one described in equation (1)).

\[
[K] = \sum_{l=1}^{n} \sum_{j=1}^{m} w_j^l [K_j^l] \quad \text{and} \quad [M] = \sum_{l=1}^{n} \sum_{j=1}^{m} w_j^l [M_j^l]
\] (38)

Where \( w \) is the weight function, \( l \) is the layer number and \( j \) is the orientation number.

With these matrixes, eigenvalues and eigenvectors can be easily obtained using Matlab’s `eigs()` function. The eigenvector is then mass normalized, as that is one of the assumptions made when the derivative’s expression (21) was deduced:

Mass normalizing is a simple matter of executing two steps:

\[
\phi^T [M] \phi = A \iff \phi^T [M] \phi = 1
\] (39)  (40)

Therefore:

\[
\bar{\phi} = \frac{\phi}{\sqrt{A}}
\] (41)
Short of entering the optimizer, only one final step remains in the optimization process: the derivative calculation. The derivatives are dependent on the number of layers and orientations that are selected before the program starts. These can lead to a large number of variables which, coupled with DMO’s equation, makes for a very long expression which makes manual deduction tricky. This can be easily understood by simple observation of equations (3) and (21).

The solution was to use Matlab’s symbolic capabilities, which allow it to calculate the derivatives by itself, with the only user input required being the number of orientations and layers, before starting the program.

The information is then passed onto the optimizer. If convergence is achieved then the optimization cycle is ended and the program enters the post-processing phase. If on the other end, convergence is not achieved, a new set of design variables $x_j^l$ is chosen by the optimizer and new weights are calculated.

The process then repeats itself with the calculation of new matrixes $[K]$ and $[M]$, new eigenvalues and mass normalized eigenvectors and by submitting them again to the optimizer until convergence is achieved.

5.5.2.3. Post-Processing

The program finishes with an optional post-processing phase, which relies almost entirely on Ansys. Its objective is to simulate all orientation possibilities, to verify the results obtained by the optimization step. This is achieved by running Ansys in batch mode $n$ times, where $n$ is the number of possible combinations and by storing the first ten eigenvalues in a cell array.

This is a very efficient way to verify results when layer number is reduced and, for cases where only 1 or 2 layers exist, it is actually faster than the optimization program. However since the number of possibilities increases exponentially, so does the post-processing time, which leads to running times of half an hour for a 5 layer case.

Post-processing should therefore be used with caution.
5.5.3. Step by Step Description of the Program

1. Reads geometry and material configuration
2. Performs an analysis of convergence
3. Initialization of cell arrays for mass [M] and stiffness [K] matrixes

4. Collects [K] and [M] from Ansys in sparse format
5. Calculates initial weight values using the DMO formulation
6. Calculates [K] and [M] for the whole structure using the weights calculated.

7. Obtains the eigenvectors and eigenvalues from the eigs() function in Matlab
8. Mass normalization of eigenvectors
9. Calculates the necessary variable for the derivative calculation ($\phi[K]\phi$)

10. Starts the optimization variables
11. Begins the optimization loop
12. Checks the iteration number and adjusts the penalty value accordingly

13. Same steps as 6,7,8,9,10
14. Calculates the derivatives and goes back to step 13
15. End of optimization loop and final output

Figure 5-8 – Step by step description of the program
6. Testing and Analysis of the Results

In order to test the good implementation of the methods and of the Matlab code, several test problems were run. These serve a double purpose, as they also allow the identification of various shortcomings, not only of the program developed in this work, but also of the theories that are behind it, particularly DMO5.

Although more cases can be found annexed to this thesis, a couple of interesting ones have been selected and will be described and analysed. These should be enough to prove not only the correct functioning of the developed optimizer, but also to show some of the phenomenon that can affect its performance, such as mode switching or DMO's tendency, in special cases, to combine both 45° and -45° orientations to create a fictitious X pattern.

Unless stated otherwise, the test cases were implemented with the penalization scheme described in condition (29) and all weight values were initialized as 0.25 as there are 4 possible fibre orientations.

Also of importance is the fact the derivative's values were normalized so that their maximum value is close to 1, a simple rescaling of the derivatives. This may seem like a trivial step, but proved itself to be of the utmost importance in some examples, to obtain the best possible optimization performance, as the optimizer seems to function best when such rescaling is applied.

6.1. Material Properties

The test cases analysed in this section, as well as those that are annexed to this work have been executed using orthotropic material properties. To the knowledge of the author, these do not correspond to any specific existing material as the examples tested are theoretical.

<table>
<thead>
<tr>
<th>$E_{11}(GPa)$</th>
<th>$E_{22}(GPa)$</th>
<th>$E_{33}(GPa)$</th>
<th>$\nu_{12}$</th>
<th>$\nu_{23}$</th>
<th>$\nu_{13}$</th>
<th>$G_{12}(GPa)$</th>
<th>$G_{23}(GPa)$</th>
<th>$G_{13}(GPa)$</th>
<th>$\rho \ (Kg/m^3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>54</td>
<td>18</td>
<td>18</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>9</td>
<td>3.4</td>
<td>9</td>
<td>1900</td>
</tr>
</tbody>
</table>

These characteristics are an attempt to simulate those of a fibre reinforced composite material, with orthotropic behaviour. The values selected should prove sufficiently different to allow the successful testing of the developed program, as well as the efficiency of the methods that are behind it. Different stacking orders should be obtained for cases with different geometry and/or boundary conditions.
6.2. Test Case 1

The first example has a simple geometry and constraints and so that the obtained results are simple to understand. It features a rectangular plate only clamped at the left hand boundary.

![Geometry and boundary conditions of test case 1](image)

From observation of the case in (Figure 6-1), one may expect the ideal orientation of the layers to be 0°. A longitudinal orientation of the fibres is therefore the optimal way of guaranteeing the increase of the component’s stiffness, since transversal fibres will add little to the overall mechanical stiffness.

The penalization used in this scheme is detailed in condition (29).

Table 2 contains the information regarding the obtained final fundamental frequency values in Hertz, the stacking order and computational time for both the final optimization result and the absolute maximum value obtained from testing all possible combinations with Ansys and thus identifying a global optimum. The number of iterations of the optimization is also listed.

Table 2 – Main results of test case 1

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Number of Iterations</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21.636</td>
<td>0°</td>
<td>18</td>
<td>41.299</td>
</tr>
<tr>
<td>2</td>
<td>43.257</td>
<td>0°/0°</td>
<td>18</td>
<td>81.099</td>
</tr>
<tr>
<td>3</td>
<td>64.851</td>
<td>0°/0°/0°</td>
<td>19</td>
<td>97.800</td>
</tr>
<tr>
<td>4</td>
<td>86.405</td>
<td>0°/0°/0°/0°</td>
<td>18</td>
<td>124.183</td>
</tr>
<tr>
<td>5</td>
<td>107.908</td>
<td>0°/0°/0°/0°/0°</td>
<td>32</td>
<td>204.585</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Error (%)</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21.638</td>
<td>0°</td>
<td>0.009</td>
<td>9.974</td>
</tr>
<tr>
<td>2</td>
<td>43.269</td>
<td>0°/0°</td>
<td>0.028</td>
<td>40.965</td>
</tr>
<tr>
<td>3</td>
<td>64.888</td>
<td>0°/0°/0°</td>
<td>0.057</td>
<td>149.486</td>
</tr>
<tr>
<td>4</td>
<td>86.489</td>
<td>0°/0°/0°/0°</td>
<td>0.097</td>
<td>633.991</td>
</tr>
<tr>
<td>5</td>
<td>108.067</td>
<td>0°/0°/0°/0°/0°</td>
<td>0.147</td>
<td>3188.529</td>
</tr>
</tbody>
</table>
Results show that the optimization achieved global minimum in all cases, as the stacking order is the same for both the Matlab results and the best result from all the possible combinations.

Also of interest is the difference between computation times. These were obtained in the same computer, which was also used for all the test cases but its value will obviously depend on the system’s processing capabilities. They are however, useful for comparing the optimization process’ efficiency when compared to the post-processor.

This computation time increases, in both cases, with the number of layers as expected, since the number of design variables increases. While it is more efficient to simply test all possible combinations in Ansys when the number of layers is inferior to 3, the optimization process’ utility becomes clear by comparing the time required to obtain results by the two methods when layer number is superior.

The number of possible combinations of fibre orientation in the composite material is given by:

\[ \text{Combinations} = A^n \]  

(42)

Where \( A \) is the number of possible fibre orientations and \( n \) is the number of layers.

If \( A = 4 \) then the possible number of combinations increases in the following fashion: \( 4, 16, 64, 256, 1024, \ldots, \infty \). As the number of possibilities increases exponentially, so does the time required for testing all of them in Ansys, rendering this method very inefficient for cases which involve many layers. Thus, in this test case, we go from a case where computation time was 4.141 times lower than the optimization time to one where it is 15.585 times higher.

The error displayed in Table 2, Table 4, Table 8, Table 10 and Table 13 refers to the difference between the fundamental frequency obtained through optimization and post-processing. It is calculated using expression (43).

\[ \text{Error} = \left( \frac{\omega_{\text{post-processing}} - \omega_{\text{optimization}}}{\omega_{\text{post-processing}}} \right) \times 100 \]

(43)

When small error values are obtained, the difference can be explained by the usage of different solvers for the calculation of the results, as the optimizer uses Matlab’s \texttt{eigs()} function and the post-processor uses Ansys. Larger error values (above 1%) can be due to a failure to obtain fully converged results of the design variables or due to more serious problems (see test case 3).

These test cases’ design variables showed great convergence to the limit values 0 and 1 (see Table 3). The number of weight functions is obtained by the multiplication of the possible fibre orientations by the number of layers. The possible orientations were set in the order of \([0^\circ, 90^\circ, 45^\circ, -45^\circ]\) or \(w_1, w_2, w_3, w_4\) for the first layer, \(w_5, w_6, w_7, w_8\) for the second one and so on.
Table 3 – Weight functions’ values for test case 1

<table>
<thead>
<tr>
<th>Weight Variables</th>
<th>1 Layer</th>
<th>2 Layers</th>
<th>3 Layers</th>
<th>4 Layers</th>
<th>5 Layers</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_1$</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>$w_2$</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_3$</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_4$</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_5$</td>
<td></td>
<td></td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>$w_6$</td>
<td></td>
<td></td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_7$</td>
<td></td>
<td></td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_8$</td>
<td></td>
<td></td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_9$</td>
<td></td>
<td></td>
<td></td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>$w_{10}$</td>
<td></td>
<td></td>
<td></td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_{11}$</td>
<td></td>
<td></td>
<td></td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_{12}$</td>
<td></td>
<td></td>
<td></td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_{13}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.000</td>
</tr>
<tr>
<td>$w_{14}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.000</td>
</tr>
<tr>
<td>$w_{15}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.000</td>
</tr>
<tr>
<td>$w_{16}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.000</td>
</tr>
<tr>
<td>$w_{17}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$w_{18}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$w_{19}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$w_{20}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Regarding the number of iterations required by the optimizer to solve the problem, no discernible pattern can be seen. The most that can be said is that the higher the layer number, the larger the number of iterations can be.

In (Figure 6-2) one can observe the natural frequency’s evolution along the optimization process. It evidences the fact that most of the iterations occur when the result is already somewhat stabilized and that the optimal results can be obtained with much less iterations should the stopping criteria be loosened.

![Natural Frequency Evolution](image)

**Figure 6-2 – Graphical display of the evolution of the natural frequency throughout the optimization – Test case 1**
6.3. Test Case 2

The second test case is an octagon with equal length and height which is clamped on 6 edges. It should show the program’s capacity to handle slightly more complex geometries.

![Figure 6-3 – Geometry and boundary conditions of test case 2](image)

The most important data about the problem has been summarized in Table 4.

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Number of Iterations</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>185.446</td>
<td>0º</td>
<td>10</td>
<td>88.371</td>
</tr>
<tr>
<td>2</td>
<td>368.892</td>
<td>0º/0º</td>
<td>7</td>
<td>133.783</td>
</tr>
<tr>
<td>3</td>
<td>549.218</td>
<td>0º/0º/0º</td>
<td>24</td>
<td>210.592</td>
</tr>
<tr>
<td>4</td>
<td>725.661</td>
<td>0º/0º/0º/0º</td>
<td>15</td>
<td>260.127</td>
</tr>
<tr>
<td>5</td>
<td>897.535</td>
<td>0º/0º/0º/0º/0º</td>
<td>28</td>
<td>372.951</td>
</tr>
</tbody>
</table>

As in test case 1, the computational time increases with the number of layers. Since this case features a higher number of elements, this effect is much more pronounced on the combination...
calculation in Ansys. As a result, the computational time required for the optimization and for testing all possible combinations has increased, while preserving a relation similar to the one seen in test case 1 that goes from the optimization being 6.611 times slower with 1 layer to it being 15.122 times faster with 5 layers.

Stacking results are also the same, the best solution being the horizontal direction in all layers. While in test case 1 the design and weight variables drifted very quickly to the limit values, in test case 2 there is a certain difficulty in the 5 layer case to drive the middle layer’s (layer 3) values \( w_9, w_{10} \) to its limit values.

<table>
<thead>
<tr>
<th>Weight Variables</th>
<th>1 Layer</th>
<th>2 Layers</th>
<th>3 Layers</th>
<th>4 Layers</th>
<th>5 Layers</th>
</tr>
</thead>
<tbody>
<tr>
<td>( w_1 )</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.995</td>
</tr>
<tr>
<td>( w_2 )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.005</td>
</tr>
<tr>
<td>( w_3 )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>( w_4 )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>( w_5 )</td>
<td>1.000</td>
<td>0.996</td>
<td>1.000</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>( w_6 )</td>
<td>0.000</td>
<td>0.004</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>( w_7 )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>( w_8 )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>( w_9 )</td>
<td>1.000</td>
<td>1.000</td>
<td>0.706</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_{10} )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.279</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_{11} )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.007</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_{12} )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.008</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_{13} )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_{14} )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_{15} )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_{16} )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_{17} )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.995</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_{18} )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.005</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_{19} )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_{20} )</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In Table 5, highlighted in bold, are the values of \( w_9 \) and \( w_{10} \), which represent the orientations 0° and 90° of the fibres on the middle layer for the 5 layered case. Unlike all the other values contained in Table 5 these are neither \( \approx 1 \) or \( \approx 0 \), instead presenting more intermediate values of 0.706 and 0.279 respectively. While they still point to 0° being the best solution, they may induce a measure of doubt as to if it is indeed the best solution.

As stated in [6], the DMO5 is more likely to yield “grey” results. Despite this knowledge, it is nonetheless important to try to understand why this is happening in this specific case. One of the possible reasons is the relative lack of impact of the stiffness properties of the middle layer as this is the one that contains the mid plane surface.
It is common knowledge that the closer a surface is to its reference plane, the smaller its contribution to the overall stiffness is and layers in composite materials are no exception. In fact the difference is very small as can be seen by testing 4 cases in which the middle layer takes one of the four orientation possibilities and all other layers have 0º value.

These are:

Table 6 – Difference between orientations in layer 3

<table>
<thead>
<tr>
<th>Stacking Order</th>
<th>Natural Frequency (Hz)</th>
<th>Difference to 0º (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0º/0º/0º/0º/0º</td>
<td>911.108</td>
<td></td>
</tr>
<tr>
<td>0º/0º/90º/0º/0º</td>
<td>906.925</td>
<td>0.459</td>
</tr>
<tr>
<td>0º/0º/45º/0º/0º</td>
<td>908.708</td>
<td>0.263</td>
</tr>
<tr>
<td>0º/0º/-45º/0º/0º</td>
<td>907.686</td>
<td>0.376</td>
</tr>
</tbody>
</table>

Note that in Table 6, the natural frequencies of the 45º and -45º cases are different. This can be explained by the irregularity of the mesh which can affect the results and, in this case, originate asymmetric results (see Figure 6-4).

Figure 6-4 – Irregular mesh obtained in test case 2.

Because the difference from the best orientation (0º) and the worst (90º) is only 0.459% the derivatives of layer three’s variables will have values several orders of magnitude inferior to the ones in the outside layers. If they are too low, the optimizer may have the tendency to make
very small increments to these design variables and therefore not be able push them to 0 or 1, before the stopping criteria ends the optimization process.

Table 7 – Design variables and derivatives on the third iteration

<table>
<thead>
<tr>
<th>Weight Variables</th>
<th>Values</th>
<th>Derivatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0.831</td>
<td>-0.004</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0.092</td>
<td>0.003</td>
</tr>
<tr>
<td>$x_3$</td>
<td>0.068</td>
<td>0.006</td>
</tr>
<tr>
<td>$x_4$</td>
<td>0.070</td>
<td>0.006</td>
</tr>
<tr>
<td>$x_5$</td>
<td>0.432</td>
<td>-0.025</td>
</tr>
<tr>
<td>$x_6$</td>
<td>0.199</td>
<td>0.004</td>
</tr>
<tr>
<td>$x_7$</td>
<td>0.195</td>
<td>0.022</td>
</tr>
<tr>
<td>$x_8$</td>
<td>0.191</td>
<td>0.022</td>
</tr>
<tr>
<td>$x_9$</td>
<td>0.264</td>
<td>-0.003</td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>0.247</td>
<td>-0.001</td>
</tr>
<tr>
<td>$x_{11}$</td>
<td>0.247</td>
<td>0.002</td>
</tr>
<tr>
<td>$x_{12}$</td>
<td>0.246</td>
<td>0.002</td>
</tr>
<tr>
<td>$x_{13}$</td>
<td>0.432</td>
<td>-0.025</td>
</tr>
<tr>
<td>$x_{14}$</td>
<td>0.199</td>
<td>0.004</td>
</tr>
<tr>
<td>$x_{15}$</td>
<td>0.195</td>
<td>0.022</td>
</tr>
<tr>
<td>$x_{16}$</td>
<td>0.191</td>
<td>0.022</td>
</tr>
<tr>
<td>$x_{17}$</td>
<td>0.831</td>
<td>-0.004</td>
</tr>
<tr>
<td>$x_{18}$</td>
<td>0.092</td>
<td>0.003</td>
</tr>
<tr>
<td>$x_{19}$</td>
<td>0.068</td>
<td>0.006</td>
</tr>
<tr>
<td>$x_{20}$</td>
<td>0.070</td>
<td>0.006</td>
</tr>
</tbody>
</table>

Table 7 shows that the further into the middle layer 3 one looks, the more intermediary the values are. It therefore seems to corroborate the theory outlined (Table 7 displays the design variables of the 0° orientation in bold).

As was the case in the first test, the natural frequency variation happens mostly in the first few iterations, with the remaining showing very little variation in the objective function.
Figure 6-5 - Graphical display of the evolution of the natural frequency throughout the optimization – Test case 2

6.4. Test Case 3

Test case 3 consists of a simply supported square plate and was selected for being the perfect geometry to show a particular deficiency of DMO5.

Figure 6-6 - Geometry and boundary conditions of test case 3
In this case, due to geometrical symmetry, layers with fibres oriented along the 45° and -45° directions have the same mechanical properties. This leads to multiple optimal solutions, something that is apparent in Table 8. This poses a particular challenge to the optimizer as it does not distinguish between the multiple solutions.

Table 8 - Main results of test case 3

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Number of Iterations</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>99.941</td>
<td>+ - 45°</td>
<td>9</td>
<td>33.391</td>
</tr>
<tr>
<td>2</td>
<td>199.258</td>
<td>+ - 45°/+ - 45°</td>
<td>9</td>
<td>54.709</td>
</tr>
<tr>
<td>3</td>
<td>297.405</td>
<td>+ - 45°/+ - 45°/+ - 45°</td>
<td>9</td>
<td>74.618</td>
</tr>
<tr>
<td>4</td>
<td>393.987</td>
<td>+ - 45°/+ - 45°/+ - 45°/+ - 45°</td>
<td>9</td>
<td>95.453</td>
</tr>
<tr>
<td>5</td>
<td>488.734</td>
<td>+ - 45°/+ - 45°/+ - 45°/+ - 45°/+ - 45°</td>
<td>33</td>
<td>222.877</td>
</tr>
</tbody>
</table>

Global Maximum From All Combinations (Ansys)

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Error (%)</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>95.093</td>
<td>+ - 45°</td>
<td>4.851</td>
<td>6.618</td>
</tr>
<tr>
<td>2</td>
<td>189.841</td>
<td>+ - 45°/+ - 45°</td>
<td>4.726</td>
<td>29.915</td>
</tr>
<tr>
<td>3</td>
<td>285.699</td>
<td>45/-45/45 or -45/45/45</td>
<td>3.936</td>
<td>115.078</td>
</tr>
<tr>
<td>4</td>
<td>386.014</td>
<td>45/-45/45/45/-45</td>
<td>2.024</td>
<td>517.932</td>
</tr>
<tr>
<td>5</td>
<td>476.816</td>
<td>45/-45/45/45/45 or 45/-45/45/45/45 or -45/45/45/45/-45/45 or -45/45/-45/45/45</td>
<td>2.439</td>
<td>2196.467</td>
</tr>
</tbody>
</table>

The existence of multiple solutions which involve both 45° and -45°, due to the nature of the optimization scheme, leads the optimizer to an intermediate “grey” solution, where the design variables that correspond to these orientations are 0.5.

Although this solution indicates that both variables define optimal orientations, they nevertheless defeat the DMO’s objective of obtaining null or unitary design variables.

When comparing the errors between the optimal solutions obtained from trying all possibilities in Ansys and the one obtained from optimization, one finds that they displays a value that is much higher than 1%. This is inconsistent with the values found in test cases 1 and 2 and therefore requires a detailed explanation.
By setting two design variables to 0.5, the optimizer is effectively creating a new and unpredicted fibre orientation. By combining both 45° and -45° fibres, one obtains a crossed pattern such as the one in Figure 6-7.

![Figure 6-7 – Crossed pattern obtained in test case 3.](image)

This pattern has slightly superior mechanical characteristics, which lead to the optimization’s results being larger than the ones obtained by Ansys optimal solutions. This is something that did not happen in the previous examples and is indeed the reason behind the unusually large errors found in this example.

The weight variables’ values can be found in Table 9.

<table>
<thead>
<tr>
<th>Weight Variables</th>
<th>1 Layer</th>
<th>2 Layers</th>
<th>3 Layers</th>
<th>4 Layers</th>
<th>5 Layers</th>
</tr>
</thead>
<tbody>
<tr>
<td>w_1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>w_2</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>w_3</td>
<td>0.500</td>
<td>0.500</td>
<td>0.500</td>
<td>0.500</td>
<td>0.500</td>
</tr>
<tr>
<td>w_4</td>
<td>0.500</td>
<td>0.500</td>
<td>0.500</td>
<td>0.500</td>
<td>0.500</td>
</tr>
<tr>
<td>w_5</td>
<td>--------</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>w_6</td>
<td>--------</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>w_7</td>
<td>--------</td>
<td>0.500</td>
<td>0.500</td>
<td>0.500</td>
<td>0.500</td>
</tr>
<tr>
<td>w_8</td>
<td>--------</td>
<td>0.500</td>
<td>0.500</td>
<td>0.500</td>
<td>0.500</td>
</tr>
<tr>
<td>w_9</td>
<td>--------</td>
<td>--------</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>w_10</td>
<td>--------</td>
<td>--------</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>w_11</td>
<td>--------</td>
<td>--------</td>
<td>0.500</td>
<td>0.500</td>
<td>0.500</td>
</tr>
<tr>
<td>w_12</td>
<td>--------</td>
<td>--------</td>
<td>0.500</td>
<td>0.500</td>
<td>0.500</td>
</tr>
<tr>
<td>w_13</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>w_14</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>w_15</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>0.500</td>
<td>0.500</td>
</tr>
<tr>
<td>w_16</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>0.500</td>
<td>0.500</td>
</tr>
<tr>
<td>w_17</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>0.000</td>
</tr>
<tr>
<td>w_18</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>0.000</td>
</tr>
<tr>
<td>w_19</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>0.500</td>
<td>0.500</td>
</tr>
<tr>
<td>w_20</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>0.500</td>
</tr>
</tbody>
</table>
With the exception of the five layered case, this example displays a remarkably small number of iterations executed before convergence was achieved. By observing Figure 6-8, one can see the typical flat lines found after a small number of iterations,

![Natural Frequency Evolution](image)

*Figure 6-8 - Graphical display of the evolution of the natural frequency throughout the optimization – Test case 3*

### 6.5. Test Case 4

In test case 4 the rectangular shape is revisited, but this time, simply supported in two consecutive sides. This example was selected as it shows an interesting fibre orientation pattern.

![Geometry and boundary conditions of test case 4](image)

*Figure 6-9 - Geometry and boundary conditions of test case 4*

Perhaps due to the more interesting stacking solutions obtained, the optimizer showed some difficulty in converging, when the penalization scheme in equation (29) was used. In order to obtain better results, a simpler one was employed:
\[
\begin{align*}
    p &= 2 \text{ when } \text{iteration} \in [1,5] \\
    p &= 3 \text{ when } \text{iteration} \in [5, \infty[ 
\end{align*}
\]

After adopting this new penalization scheme, more accurate results were obtained. These are displayed in Table 10.

**Table 10 - Main results of test case 4**

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Number of Iterations</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>37.981</td>
<td>45º</td>
<td>10</td>
<td>44.428</td>
</tr>
<tr>
<td>2</td>
<td>75.526</td>
<td>45º/45º</td>
<td>14</td>
<td>78.598</td>
</tr>
<tr>
<td>3</td>
<td>112.332</td>
<td>45º/-45º/45º</td>
<td>12</td>
<td>101.870</td>
</tr>
<tr>
<td>4</td>
<td>146.917</td>
<td>45º/-45º/45º</td>
<td>8</td>
<td>115.921</td>
</tr>
<tr>
<td>5</td>
<td>183.104</td>
<td>45º/-45º/45º/45º</td>
<td>15</td>
<td>243.749</td>
</tr>
</tbody>
</table>

**Global Maximum From All Combinations (Ansys)**

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Error (%)</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>36.769</td>
<td>45º</td>
<td>3.191</td>
<td>9.248</td>
</tr>
<tr>
<td>2</td>
<td>73.294</td>
<td>45º/45º</td>
<td>2.955</td>
<td>36.669</td>
</tr>
<tr>
<td>3</td>
<td>110.102</td>
<td>45º/-45º/45º</td>
<td>1.985</td>
<td>167.581</td>
</tr>
<tr>
<td>4</td>
<td>147.752</td>
<td>45º/-45º/-45º/45º</td>
<td>0.565</td>
<td>765.571</td>
</tr>
<tr>
<td>5</td>
<td>184.566</td>
<td>45º/-45º/45º/-45º/45º</td>
<td>0.792</td>
<td>3203.181</td>
</tr>
</tbody>
</table>

From Table 10, one can observe a peculiar evolution of the error obtained when comparing the optimization result with the absolute maximum. While in previous examples the error tended to gradually increase with the number of layers, in this test case, the error has its maximum value in the 1 layer case with a large value of 3.191%. This then decreases as the number of layers is increased to 4, where error is a much more reasonable 0.565%.
To understand the reason behind this one must look into the weight function’s values, which contain the explanation of this behaviour.

Table 11 - Weight functions’ values for test case 4

<table>
<thead>
<tr>
<th>Weight Variables</th>
<th>1 Layer</th>
<th>2 Layers</th>
<th>3 Layers</th>
<th>4 Layers</th>
<th>5 Layers</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_1$</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_2$</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_3$</td>
<td>0.710</td>
<td>0.711</td>
<td>0.727</td>
<td>0.998</td>
<td>1.000</td>
</tr>
<tr>
<td>$w_4$</td>
<td>0.290</td>
<td>0.288</td>
<td>0.273</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_5$</td>
<td>--------</td>
<td>0.000</td>
<td>0.004</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_6$</td>
<td>--------</td>
<td>0.000</td>
<td>0.004</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_7$</td>
<td>--------</td>
<td>0.711</td>
<td>0.012</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_8$</td>
<td>--------</td>
<td>0.288</td>
<td>0.981</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>$w_9$</td>
<td>--------</td>
<td>--------</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_{10}$</td>
<td>--------</td>
<td>--------</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_{11}$</td>
<td>--------</td>
<td>--------</td>
<td>0.752</td>
<td>0.000</td>
<td>0.012</td>
</tr>
<tr>
<td>$w_{12}$</td>
<td>--------</td>
<td>--------</td>
<td>0.248</td>
<td>1.000</td>
<td>0.988</td>
</tr>
<tr>
<td>$w_{13}$</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_{14}$</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_{15}$</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>0.998</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_{16}$</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>0.000</td>
<td>1.000</td>
</tr>
<tr>
<td>$w_{17}$</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_{18}$</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_{19}$</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>1.000</td>
</tr>
<tr>
<td>$w_{20}$</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>0.000</td>
</tr>
</tbody>
</table>

From Table 11 the cause of the result’s errors can be devised by analysing the 1, 2 or 3 layered cases. In these 3 particular optimization problems, the optimizer has difficulty in pushing the
design variables and therefore the weight variables, to their limit values. This leads to a solution where one finds “grey” results such as $w_3 = 0.710$ or $w_4 = 0.290$.

These intermediate values, like in test case 3, lead to a crossed fibre structure whose mechanical properties are superior to either of their individual contributors. This explains why the natural frequency obtained in Matlab is superior to the one obtained from Ansys and it also explains the high error values.

Once the weights start taking unitary or null values in the 4 or 5 layered cases the error decreases to acceptable values.

Aside from the aforementioned problems, the best solution found by the optimizer for a case with 5 layers is the stacking order of 45°/-45°/-45°/-45°/45°. By testing all possible solutions, one finds the global maximum is obtained by stacking layers with fibre orientations of 45°/-45°/45°/-45°/45°. This means that the optimizer arrived at a local minimum point instead. The difference in frequencies of both stacking orders is however, very small, as the natural frequencies of both orders take the values in (Table 12) when tested in Ansys.

\begin{table}[h]
\centering
\caption{Difference between the natural frequencies obtained with the stacking orders of the global maximum and the local maximum obtained by the optimizer.}
\begin{tabular}{|c|c|}
\hline
Stacking Order & Natural Frequency (Hz) \\
\hline
[45°/-45°/45°/-45°/45°] & 184.566 \\
\hline
[45°/-45°/-45°/-45°/45°] & 184.468 \\
\hline
\end{tabular}
\end{table}

Despite the fact that the difference is so small as to be negligible, one should keep in mind that the optimization process can still arrive at a local minimum case.

Despite the difficult convergence of the solution, it nevertheless shows the optimizer’s ability to arrive at complex stacking solutions, where fibre orientation changes from layer to layer.

On cases with 3 or more layers, it becomes beneficial to switch fibre orientation from 45° to -45° on consecutive layers. In order to understand the reason behind this one should look at Figure 6-11.

A fibre is at its stiffest when connecting two supported sides, something which favours the 45° orientation and that can be verified by the results in cases with 1 or 2 layers. However, by using a 45° orientation, an entire section of the geometry will be left with fibres that only connect unsupported sides and therefore do not add greatly to overall plate stiffness. By using alternate layers with fibres at a -45° angle this no longer happens as the blue area in Figure 6-11 now has fibres with one end at the inferior side of the rectangle which is simply supported.
As with the previous examples, maximum natural frequency is obtained in a reduced number of iterations as the optimizer spends most of its iterations on the somewhat flat parts of the lines seen in Figure 6-12.

**Figure 6-12** - Graphical display of the evolution of the natural frequency throughout the optimization – Test case 4
6.6. Test Case 5

The last test case featured consists of a simply supported triangle. This particular case provides a good idea of what happens when mode switching occurs.

![Figure 6-13 - Geometry and boundary conditions of test case 5](image)

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Number of Iterations</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>249.802</td>
<td>45°</td>
<td>8</td>
<td>118.192</td>
</tr>
<tr>
<td>2</td>
<td>496.633</td>
<td>45°/45°</td>
<td>10</td>
<td>154.247</td>
</tr>
<tr>
<td>3</td>
<td>738.438</td>
<td>45°/45°/45°</td>
<td>17</td>
<td>192.946</td>
</tr>
<tr>
<td>4</td>
<td>958.878</td>
<td>45°/-45°/-45°/45°</td>
<td>27</td>
<td>269.148</td>
</tr>
<tr>
<td>5</td>
<td>--------------------------</td>
<td>?/?/?/?/?/?</td>
<td>-------------------</td>
<td>----------------------</td>
</tr>
</tbody>
</table>

Global Maximum From All Combinations (Ansys)

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Error (%)</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>250.052</td>
<td>45°</td>
<td>0.100</td>
<td>9.206</td>
</tr>
<tr>
<td>2</td>
<td>498.194</td>
<td>45°/45°</td>
<td>0.313</td>
<td>35.980</td>
</tr>
<tr>
<td>3</td>
<td>742.935</td>
<td>45°/45°/45°</td>
<td>0.605</td>
<td>167.466</td>
</tr>
<tr>
<td>4</td>
<td>964.369</td>
<td>45°/-45°/-45°/45°</td>
<td>0.569</td>
<td>758.844</td>
</tr>
<tr>
<td>5</td>
<td>1030.135</td>
<td>90°/-45°/-45°/45°/-45°/45° or 45°/-45°/-45°/45°/90°</td>
<td>----------------</td>
<td>3283.589</td>
</tr>
</tbody>
</table>
Due to mode switching, this particular geometry and boundary conditions can only be solved when the number of layers is inferior to 5. Cases with higher number of layers proved impossible to optimize as the application fails to converge.

To better understand why this happens we need to look into the vibration frequencies obtained during optimization, specifically, to the first two frequencies obtained in the first and last iterations.

Table 14 – First and second frequencies on the first and last iterations of optimization

<table>
<thead>
<tr>
<th>Layers</th>
<th>Natural Frequency (Hz)</th>
<th>Second Frequency (Hz)</th>
<th>Difference between frequencies (%)</th>
<th>Natural Frequency (Hz)</th>
<th>Second Frequency (Hz)</th>
<th>Difference between frequencies (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Layer</td>
<td>218.787</td>
<td>494.084</td>
<td>55.719</td>
<td>249.802</td>
<td>502.309</td>
<td>50.269</td>
</tr>
<tr>
<td>2 Layers</td>
<td>435.729</td>
<td>970.707</td>
<td>55.112</td>
<td>496.633</td>
<td>777.985</td>
<td>36.164</td>
</tr>
<tr>
<td>3 Layers</td>
<td>649.462</td>
<td>970.744</td>
<td>33.096</td>
<td>738.438</td>
<td>778.019</td>
<td>5.087</td>
</tr>
<tr>
<td>4 Layers</td>
<td>859.230</td>
<td>970.838</td>
<td>11.496</td>
<td>958.879</td>
<td>958.958</td>
<td>0.008</td>
</tr>
<tr>
<td>5 Layers</td>
<td>971.033</td>
<td>1064.467</td>
<td>8.778</td>
<td>1001.709</td>
<td>1034.200</td>
<td>3.142</td>
</tr>
</tbody>
</table>

Table 14 contains very interesting information. Firstly, one can clearly see that increasing the number of layers does not affect all frequencies in the same way, as the first natural frequency always increases, but the second frequency can either increase or remain almost the same, depending on the number of layers.

Secondly, one can readily observe that the higher the number of layers, the closer together the first and second frequencies are to each other, at the beginning of the optimization process. This can explain the apparent difficulty in solving the 5 and 4 layered problems, as the proximity between the first 2 modes can potentiate mode switching. If mode switching happens then the eigenvectors will switch, causing radical change of the derivatives and therefore, of the search direction.

Despite this phenomenon, the 4 layered case was still successfully solved, albeit with a very large number of iterations, perhaps due to the larger initial difference.
To better observe the evolution of the first two natural frequencies, these values were collected and introduced in Figure 6-15. From its observation one can conclude that as the optimization goes on, the difference between these frequencies steadily reduces until becoming negligible. There are also two sections of the graphic that display irregularities, as the frequencies increase and decrease from one iteration to the next due to mode switching.

In order to cross check the theory that the optimization failed on the 5 layered case due to mode switching, we need to look into some of the derivatives obtained in 2 consecutive iterations.
Table 15 – Derivative’s values on the first two iterations of problem 5 with 5 layers.

<table>
<thead>
<tr>
<th>Derivatives</th>
<th>First iteration</th>
<th>Second iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1$</td>
<td>0.068</td>
<td>-0.070</td>
</tr>
<tr>
<td>$d_2$</td>
<td>-0.009</td>
<td>-0.117</td>
</tr>
<tr>
<td>$d_3$</td>
<td>0.301</td>
<td>-0.320</td>
</tr>
<tr>
<td>$d_4$</td>
<td>-0.330</td>
<td>0.221</td>
</tr>
<tr>
<td>$d_5$</td>
<td>0.068</td>
<td>-0.019</td>
</tr>
<tr>
<td>$d_6$</td>
<td>-0.009</td>
<td>-0.031</td>
</tr>
<tr>
<td>$d_7$</td>
<td>0.301</td>
<td>-0.085</td>
</tr>
<tr>
<td>$d_8$</td>
<td>-0.330</td>
<td>0.059</td>
</tr>
<tr>
<td>$d_9$</td>
<td>0.068</td>
<td>-0.001</td>
</tr>
<tr>
<td>$d_{10}$</td>
<td>-0.009</td>
<td>-0.002</td>
</tr>
<tr>
<td>$d_{11}$</td>
<td>0.301</td>
<td>-0.007</td>
</tr>
<tr>
<td>$d_{12}$</td>
<td>-0.330</td>
<td>0.005</td>
</tr>
<tr>
<td>$d_{13}$</td>
<td>0.068</td>
<td>-0.019</td>
</tr>
<tr>
<td>$d_{14}$</td>
<td>-0.009</td>
<td>-0.031</td>
</tr>
<tr>
<td>$d_{15}$</td>
<td>0.301</td>
<td>-0.085</td>
</tr>
<tr>
<td>$d_{16}$</td>
<td>-0.330</td>
<td>0.059</td>
</tr>
<tr>
<td>$d_{17}$</td>
<td>0.068</td>
<td>-0.070</td>
</tr>
<tr>
<td>$d_{18}$</td>
<td>-0.009</td>
<td>-0.117</td>
</tr>
<tr>
<td>$d_{19}$</td>
<td>0.301</td>
<td>-0.320</td>
</tr>
<tr>
<td>$d_{20}$</td>
<td>-0.330</td>
<td>0.221</td>
</tr>
</tbody>
</table>

Table 15 illustrates perfectly the difficulties to the optimization problem that can arise from mode switching. In it, one finds great difference between the values in two consecutive iterations, with some derivatives switching signal, like for example the third derivative. This will cause the search direction to radically change and will likely flip orientation every time an iteration is done by the optimizer.

Figure 6-16 - Evolution of the first 2 natural frequencies in the 5 layered case – Test case 5
This behaviour is perfectly illustrated in Figure 6-16, where for much of the optimization the frequencies fluctuate wildly. After the 41\textsuperscript{st} iteration this behaviour ceases and the frequencies stabilize in a non-optimal, “grey” solution; i.e. the optimizer fails.

Despite this failure, the optimizer was able to reach excellent solutions on all cases with less than five layers which can be verified by the small errors obtained (smaller than 0.61%).

Weight variables were also successfully pushed to the limit values (see Table 16) which, for the 4 layered case is an impressive achievement, considering that it was, with all probability, plagued by some mode switching problems on its last iterations.

**Table 16 - Weight functions’ values for test case 5**

<table>
<thead>
<tr>
<th>Weight Variables</th>
<th>1 Layer</th>
<th>2 Layers</th>
<th>3 Layers</th>
<th>4 Layers</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_1$</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_2$</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_3$</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.977</td>
</tr>
<tr>
<td>$w_4$</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.022</td>
</tr>
<tr>
<td>$w_5$</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>$w_6$</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.003</td>
</tr>
<tr>
<td>$w_7$</td>
<td>1.000</td>
<td>0.999</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>$w_8$</td>
<td>0.000</td>
<td>0.000</td>
<td>0.997</td>
<td></td>
</tr>
<tr>
<td>$w_9$</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>$w_{10}$</td>
<td>0.000</td>
<td>0.000</td>
<td>0.003</td>
<td></td>
</tr>
<tr>
<td>$w_{11}$</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>$w_{12}$</td>
<td>0.000</td>
<td>0.997</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$w_{13}$</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$w_{14}$</td>
<td>0.000</td>
<td>0.992</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$w_{15}$</td>
<td>0.008</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Further evidence of the effects of mode switching on the case with 4 layers, can be found by looking at the evolution of natural frequency along the iterations. Unlike any other example seen so far, the yellow line in Figure 6-17 shows many irregularities as it goes up and down on consecutive iterations, before stabilizing after a large number of iterations.
Figure 6-17 - Graphical display of the evolution of the natural frequency throughout the optimization – Test case 5
7. Conclusions and Future Work

The optimization application that was developed has been successful in its objective, the application of the discrete material optimization method, to flat plate structures for vibration criteria. It was able to yield fibre stacking orders that vastly improved the mechanical properties of the component and, for most cases, achieved the global maximum of the optimization problem.

It nevertheless shows several shortcomings that need to be addressed before a larger scale and practical use of it is undertaken. Due to the particular nature of the vibration analysis mode switching needs to be addressed by tweaking the problem formulation and therefore both the sensitivity analysis and the optimization algorithm.

DMO’s scheme 5 shows some deficiencies that are particularly evident in cases were 45º and -45º are both optimal solutions. DMO5 has a tendency to reach intermediate values for the design variables which, not only does not allow the user to know the best stacking order, but also may yield unrealistically high frequency values. This should be avoided by using a separate auxiliary penalization scheme that would act on these specific cases and allow the application to select one optimal solution.

With these “fixes” in place, the optimizer should be able to deal with most, if not all, cases with simple geometry. Since the computational time gain is remarkably large when compared to a simple iterative procedure where all possibilities are tested, it could become a very useful design tool.

To further increase the versatility of this tool, its capabilities should be expanded, to allow it to:

- Choose not only the best possible fibre orientation but also the best possible fibre material;
- Optimize cases of complex geometry such as 3-D or curved structures;

The first of these points is relatively straightforward and could be achieved with little changes to the program’s algorithm and underlying mathematical expressions, by changing the derivatives expressions to include the extra design variables and the mass matrix’s contribution, which would no longer be null.

As to the second point, it would be beneficial to approach the problem in a more compartmentalized way, namely by using information from each element. That could be achieved more efficiently by swapping the FEA program and using *Abaqus* instead of *Ansys*.
8. References


A. Additional Test Cases

This section will be dedicated to displaying additional test cases which were not analysed in the main body of this thesis.

A.1. Test Case 6

Test case 6 consists of a simply supported rectangular plate whose length is twice as large as its height. This case, like test case 5, is afflicted by mode switching and has therefore, only been tested until the 4 layered case.

![Geometry and Boundary conditions of test case 6](image)

All the most relevant data of this test case can be found in Table 17.

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Number of Iterations</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>270.608</td>
<td>90°</td>
<td>5</td>
<td>16.026</td>
</tr>
<tr>
<td>2</td>
<td>539.452</td>
<td>90°/90°</td>
<td>6</td>
<td>34.153</td>
</tr>
<tr>
<td>3</td>
<td>804.843</td>
<td>90°/90°/90°</td>
<td>8</td>
<td>53.865</td>
</tr>
<tr>
<td>4</td>
<td>1063.565</td>
<td>90°/90°/90°/90°</td>
<td>9</td>
<td>110.549</td>
</tr>
<tr>
<td>5</td>
<td>--------------</td>
<td>?/?/?/?/?</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

Global Maximum From All Combinations (Ansys)

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Error (%)</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>270.839</td>
<td>90°</td>
<td>0.085</td>
<td>6.423</td>
</tr>
<tr>
<td>2</td>
<td>539.774</td>
<td>90°/90°</td>
<td>0.060</td>
<td>27.706</td>
</tr>
<tr>
<td>3</td>
<td>808.490</td>
<td>90°/90°/90°</td>
<td>0.451</td>
<td>127.710</td>
</tr>
<tr>
<td>4</td>
<td>1071.754</td>
<td>90°/90°/90°/90°</td>
<td>0.764</td>
<td>549.502</td>
</tr>
<tr>
<td>5</td>
<td>1287.157</td>
<td>90°/45°/0°/45°/90°</td>
<td>----</td>
<td>3548.491</td>
</tr>
</tbody>
</table>
A.2. Test Case 7

Test case 7 has, like test case 6, rectangular geometry. However, boundary conditions are now limited to both sides being simply supported.

![Figure A-2 - Geometry and Boundary conditions of test case 7](image)

In this test case, no special phenomena occurs and convergence is easy to achieve, as the problem has an easy solution. Information was therefore collected successfully for any given number of layers and this data can be found in Table 18.

**Table 18 - Main results of test case 7**

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Number of Iterations</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>61.155</td>
<td>0°</td>
<td>17</td>
<td>47.097</td>
</tr>
<tr>
<td>2</td>
<td>122.235</td>
<td>0°/0°</td>
<td>9</td>
<td>60.694</td>
</tr>
<tr>
<td>3</td>
<td>183.162</td>
<td>0°/0°/0°</td>
<td>17</td>
<td>107.375</td>
</tr>
<tr>
<td>4</td>
<td>243.866</td>
<td>0°/0°/0°/0°</td>
<td>17</td>
<td>115.449</td>
</tr>
<tr>
<td>5</td>
<td>304.274</td>
<td>0°/0°/0°/0°/0°</td>
<td>37</td>
<td>212.733</td>
</tr>
</tbody>
</table>

**Global Maximum From All Combinations (Ansys)**

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Error (%)</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>61.163</td>
<td>0°</td>
<td>0.013</td>
<td>7.682</td>
</tr>
<tr>
<td>2</td>
<td>122.294</td>
<td>0°/0°</td>
<td>0.048</td>
<td>29.169</td>
</tr>
<tr>
<td>3</td>
<td>183.358</td>
<td>0°/0°/0°</td>
<td>0.107</td>
<td>128.828</td>
</tr>
<tr>
<td>4</td>
<td>244.324</td>
<td>0°/0°/0°/0°</td>
<td>0.187</td>
<td>561.941</td>
</tr>
<tr>
<td>5</td>
<td>305.160</td>
<td>0°/0°/0°/0°/0°</td>
<td>0.290</td>
<td>2414.219</td>
</tr>
</tbody>
</table>
A.3. Test Case 8

Test case 8 is very similar to test case 7, the main difference being that the simply supported sides are now the horizontal instead.

The changes to the boundary conditions makes the example susceptible to mode switching when the number of layers is higher than 4. As such, information shall not be provided for it in Table 19.

Table 19 - Main results of test case 8

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Number of Iterations</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>245.262</td>
<td>90°</td>
<td>7</td>
<td>31.452</td>
</tr>
<tr>
<td>2</td>
<td>489.359</td>
<td>90°/90°</td>
<td>13</td>
<td>63.038</td>
</tr>
<tr>
<td>3</td>
<td>731.094</td>
<td>90°/90°/90°</td>
<td>13</td>
<td>129.134</td>
</tr>
<tr>
<td>4</td>
<td>969.386</td>
<td>90°/90°/90°/90°</td>
<td>17</td>
<td>167.210</td>
</tr>
<tr>
<td>5</td>
<td>---------------------------</td>
<td>?/?/?/?/?</td>
<td>---------------------</td>
<td>-----------------------</td>
</tr>
</tbody>
</table>

Global Maximum From All Combinations (Ansys)

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Error (%)</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>245.387</td>
<td>90°</td>
<td>0.051</td>
<td>6.985</td>
</tr>
<tr>
<td>2</td>
<td>490.260</td>
<td>90°/90°</td>
<td>0.184</td>
<td>30.788</td>
</tr>
<tr>
<td>3</td>
<td>734.112</td>
<td>90°/90°/90°</td>
<td>0.411</td>
<td>123.953</td>
</tr>
<tr>
<td>4</td>
<td>976.450</td>
<td>90°/90°/90°/90°</td>
<td>0.723</td>
<td>525.832</td>
</tr>
<tr>
<td>5</td>
<td>1202.765</td>
<td>90°/90°/90°/90°</td>
<td>2179.626</td>
<td></td>
</tr>
</tbody>
</table>
A.4. Test Case 9

Still keeping with the rectangular geometry, test case 9 features fully clamped boundary conditions.

The optimization process of this geometry occurred without issues. The derivatives were rescaled to avoid the large values that appear due to this case’s high stiffness properties. Problem data is collected in Table 20.

Perhaps the biggest surprise of the results is the error obtained. This starts out low at 0.290% but quickly goes up to a maximum of 5.322% which is already a considerable difference between the results obtained from Matlab and Ansys.

Table 20 - Main results of test case 9

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Number of Iterations</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>582.478</td>
<td>90º</td>
<td>7</td>
<td>29.628</td>
</tr>
<tr>
<td>2</td>
<td>1150.882</td>
<td>90º/90º</td>
<td>10</td>
<td>59.996</td>
</tr>
<tr>
<td>3</td>
<td>1692.552</td>
<td>90º/90º/90º</td>
<td>9</td>
<td>78.265</td>
</tr>
<tr>
<td>4</td>
<td>2196.885</td>
<td>90º/90º/90º/90º</td>
<td>9</td>
<td>100.403</td>
</tr>
<tr>
<td>5</td>
<td>2661.265</td>
<td>90º/90º/90º/90º/90º</td>
<td>17</td>
<td>152.662</td>
</tr>
</tbody>
</table>

Global Maximum From All Combinations (Ansys)

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Error (%)</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>584.171</td>
<td>90º</td>
<td>0.290</td>
<td>7.347</td>
</tr>
<tr>
<td>2</td>
<td>1162.542</td>
<td>90º/90º</td>
<td>1.003</td>
<td>29.379</td>
</tr>
<tr>
<td>3</td>
<td>1729.614</td>
<td>90º/90º/90º</td>
<td>2.143</td>
<td>138.068</td>
</tr>
<tr>
<td>4</td>
<td>2280.450</td>
<td>90º/90º/90º/90º</td>
<td>3.664</td>
<td>578.592</td>
</tr>
<tr>
<td>5</td>
<td>2810.867</td>
<td>90º/90º/90º/90º/90º</td>
<td>5.322</td>
<td>2389.058</td>
</tr>
</tbody>
</table>
A.5. Test Case 10

Test case 10 is very similar to test case 7, with the difference that it is no longer simply supported. The two vertical sides are now clamped.

![Figure A-5 - Geometry and Boundary conditions of test case 10](image)

Results from this test case are very similar to the ones in the seventh example, with the exception that the plate is now much stiffer. As a result, the natural frequency is in this case much higher.

*Table 21 - Main results of test case 10*

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Number ofIterations</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>140.529</td>
<td>0°</td>
<td>7</td>
<td>38.220</td>
</tr>
<tr>
<td>2</td>
<td>280.184</td>
<td>0°/0°</td>
<td>7</td>
<td>61.616</td>
</tr>
<tr>
<td>3</td>
<td>418.116</td>
<td>0°/0°/0°</td>
<td>9</td>
<td>87.207</td>
</tr>
<tr>
<td>4</td>
<td>553.543</td>
<td>0°/0°/0°/0°</td>
<td>16</td>
<td>131.119</td>
</tr>
<tr>
<td>5</td>
<td>685.743</td>
<td>0°/0°/0°/0°/0°</td>
<td>19</td>
<td>161.390</td>
</tr>
</tbody>
</table>

*Global Maximum From All Combinations (Ansys)*

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Error (%)</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>140.624</td>
<td>0°</td>
<td>0.068</td>
<td>7.457</td>
</tr>
<tr>
<td>2</td>
<td>280.901</td>
<td>0°/0°</td>
<td>0.255</td>
<td>30.595</td>
</tr>
<tr>
<td>3</td>
<td>420.494</td>
<td>0°/0°/0°</td>
<td>0.566</td>
<td>131.940</td>
</tr>
<tr>
<td>4</td>
<td>559.075</td>
<td>0°/0°/0°/0°</td>
<td>0.989</td>
<td>638.496</td>
</tr>
<tr>
<td>5</td>
<td>696.328</td>
<td>0°/0°/0°/0°/0°</td>
<td>1.520</td>
<td>2729.740</td>
</tr>
</tbody>
</table>
A.6. Test Case 11

The rectangular geometry of this problem is clamped on 3 sides, with its remaining one free.

![Figure A-6 - Geometry and Boundary conditions of test case 11](image)

This example was solved without any difficulties, with easy convergence and very accurate results.

### Table 22 - Main results of test case 11

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Number of Iterations</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>86.685</td>
<td>90°</td>
<td>9</td>
<td>31.861</td>
</tr>
<tr>
<td>2</td>
<td>173.176</td>
<td>90°/90°</td>
<td>9</td>
<td>55.386</td>
</tr>
<tr>
<td>3</td>
<td>259.272</td>
<td>90°/90°/90°</td>
<td>18</td>
<td>92.806</td>
</tr>
<tr>
<td>4</td>
<td>344.791</td>
<td>90°/90°/90°/90°</td>
<td>13</td>
<td>101.163</td>
</tr>
<tr>
<td>5</td>
<td>429.542</td>
<td>90°/90°/90°/90°/90°</td>
<td>17</td>
<td>158.342</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Fundamental Frequency (Hz)</th>
<th>Stacking</th>
<th>Error (%)</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>86.707</td>
<td>90°</td>
<td>0.025</td>
<td>6.365</td>
</tr>
<tr>
<td>2</td>
<td>173.328</td>
<td>90°/90°</td>
<td>0.088</td>
<td>26.764</td>
</tr>
<tr>
<td>3</td>
<td>259.779</td>
<td>90°/90°/90°</td>
<td>0.195</td>
<td>114.317</td>
</tr>
<tr>
<td>4</td>
<td>345.977</td>
<td>90°/90°/90°/90°</td>
<td>0.343</td>
<td>516.059</td>
</tr>
<tr>
<td>5</td>
<td>431.841</td>
<td>90°/90°/90°/90°/90°</td>
<td>0.532</td>
<td>2109.729</td>
</tr>
</tbody>
</table>
B. User’s Manual

Welcome to the optimization program user’s manual. In this manual the user will find all the information required not only to run test cases, but also to change the material properties used, its geometry and boundary conditions. This manual is broken down into several parts for easier reading:

B.1. Creating the problem instance;
B.2. Solving;
B.3. Running the post-processing algorithm.

B.1. Creating the Problem Instance

To manipulate the problem’s characteristics, four files must be accessed:

- *Geometria.m*;
- *Variaveis.m*;
- *Txt.m*;
- *Txt_aux.m*.

The first of these files controls the problem’s geometry. In it, one finds the necessary fields for the plates length, height, for the number of starting mesh elements along x and y (minimum values), for the number of layers and their thickness.

This information is then stored in a cell array named *Geo*, as the file which is ported into the main file *Inicial.m*.

```matlab
function [Geo] = Geometria ()
Geo = cell(2,6);
Geo{1,1} = 'Comprimento';
Geo{1,2} = 'Altura';
Geo{1,3} = 'n_elementos_x';
Geo{1,4} = 'n_elementos_y';
Geo{1,6} = 'espessura_canadas';
Geo{1,6} = 'espessura_canadas';

%Variable Initialization
comp=0.2; %Length
alt=0.1; %Height
n_elementos_x=10; %Number of starting elements along x
n_elementos_y=10; %Number of starting elements along y
numero_canadas=6; %Number of layers
espessura_canadas=0.001; %Thickness of the layers

%End of variable Initialization
Geo{2,1} = comp; %Length
Geo{2,2} = alt; %Height
Geo{2,3} = n_elementos_x; %Number of starting elements along x
Geo{2,4} = n_elementos_y; %Number of starting elements along y
Geo{2,5} = numero_canadas; %Number of layers
Geo{2,6} = espessura_canadas; %Thickness of the layers
```

*Figure B-1 – Geometry parameters in Geometria.m*
The material properties can be found in the file `Variaveis.m`. All the following material properties and fibre characteristics are initialized and set in this file:

- The Young's modulus $E_{11}$, $E_{22}$, $E_{33}$;
- The Poisson's coefficients $\nu_{12}$, $\nu_{23}$, $\nu_{13}$;
- The Shear modulus $G_{12}$, $G_{23}$, $G_{13}$;
- The material's density $\rho$;
- The number of possible fibre orientations;
- The various fibre orientations accepted;
- The stacking order (Should be kept empty in this file).

All this information is ported to the main program file `Inicial.m` in two cell arrays, `Orientacoes` and `Var`. The first of these cell array files contains all the information related to the fibre orientations. The last one, `Var`, contains all the information about the material properties.

```matlab
function [Var,Orientacoes] = Variaveis ()

%Function that contains all problem properties (Materials, -%possible orientations etc)

n_materials = 1;

Var=cell(n_materials+1,7); %Creates cell array where all the material %properties are stored
Var(1,1)='EX'; %Exx Youngs modulus
Var(1,2)='Ey'; %Eyy Youngs modulus
Var(1,3)='Ez'; %Ezz Youngs modulus
Var(1,4)='Poisson12'; %Poisson12
Var(1,5)='Gxy'; %Shear modulus xy
Var(1,6)='Gyz'; %Shear modulus yz
Var(1,7)='Ro'; %Material Density
Var(1,8)='Poisson23'; %Poisson23
Var(1,9)='Poisson13'; %Poisson13

Orientacoes = cell(2,3); %Cell array where the number of possible %orientations, their angles and stacking sequence are stored
Orientacoes(1,1) = 'N direccoes'; %Number of possible fibre orientations
Orientacoes(1,2) = 'Angulos'; %Possible fibre orientations
Orientacoes(1,3) = 'Disposicoes_Camadas'; %Stacking order
```

*Figure B-2 – Cell array initialization, with all the entries descriptions.*

The information regarding material properties and fibre orientations should be introduced in the fields visible in Figure B-3 and Figure B-4 respectively.
%Material 1 – Orthotropic

\[ \begin{align*}
E_x &= 5 \times 10^9 \text{ Pa} \\
E_y &= 1 \times 10^9 \text{ Pa} \\
E_z &= E_y; & \text{ %Ex (Pa)} \\
\text{Var}(2,1) &= E_x; & \text{ %Ey (Pa)} \\
\text{Var}(2,2) &= E_y; & \text{ %Ez (Pa)} \\
\text{Var}(2,3) &= E_z; & \text{ %Ez (Pa)}
\end{align*} \]

\[ \begin{align*}
PR_{12} &= 0.2; \\
PR_{23} &= 0.2; \\
PR_{31} &= 0.2;
\end{align*} \]

\[ \begin{align*}
\text{Var}(2,4) &= PR_{12}; & \text{ %Poisson's Coefficient} \\
\text{Var}(2,5) &= PR_{23}; & \text{ %Poisson's Coefficient} \\
\text{Var}(2,6) &= PR_{31}; & \text{ %Poisson's Coefficient}
\end{align*} \]

\[ \begin{align*}
G_{xy} &= 9 \times 10^9 \text{ Pa} \\
G_{yz} &= 3.4 \times 10^9 \text{ Pa}
\end{align*} \]

\[ \begin{align*}
\text{Var}(2,7) &= G_{xy}; & \text{ %Gxy (Pa)} \\
\text{Var}(2,8) &= G_{yz}; & \text{ %Gyz (Pa)}
\end{align*} \]

\[ \begin{align*}
\rho_0 &= 1900; & \text{ %kg/m}^3
\end{align*} \]

\[ \begin{align*}
\text{Var}(2,9) &= \rho_0; & \text{ %Ro (kg/m}^3\text{)}
\end{align*} \]

Figure B-3 – Material property’s fields.

Note that there are two different materials set in this document, the first one being the composite material and the other a “null” material (Material 2) used when collecting the mass and stiffness matrixes on each layer. The second material’s properties should be kept as they are.

%Possible orientations for the fibre.

\[ \begin{align*}
n_{\text{directions}} &= 4; & \text{ %Number of possible fibre orientations.} \\
\text{Orientations}(2,1) &= n_{\text{directions}}; & \text{ %Number of possible fibre orientations}
\end{align*} \]

\[ \begin{align*}
\theta_0 &= [0 \ 90 \ 45 \ -45]; & \text{ %Possible fibre orientations 0°, 90°, 45° e -45°} \\
\text{Orientations}(2,2) &= \theta_0; & \text{ %Possible fibre orientations}
\end{align*} \]

Figure B-4 – Fibre characteristics and their respective fields.

To finalize the problem’s description, only the boundary conditions remain. These should be input in two different Matlab files, \texttt{txt.m} and \texttt{txt_aux.m}.

These two files are used to write Ansys commands to a text file used to control Ansys in batch mode. They both have similar content with the difference between them being that \texttt{txt_aux.m} contains extra commands for the output of \([K]\) and \([M]\) by Ansys.
In txt.m, the only lines of code which are used for the control of the boundary conditions are found between 126 and 132 and 255 and 261.

In lines 126-129 one finds the boundary conditions for the 4 sides of the rectangular geometry. These sides have been numbered according to Figure B-5.

These commands include a term such as ALL, UZ, UX, UY, ROTX, ROTY or ROTZ. These commands refer to the degrees of freedom that are constrained on the particular side of the rectangle (U pertains to displacements and ROT to rotations). Generally, the most used of these are the ALL and UZ command. When ALL is used, all degrees of freedom are constrained and the side is clamped. UZ refers to the simply supported condition.

Commands 48, 48_1 and 49 are additional constraints that are applied to two of the nodes of the FEA analysis to ensure that all degrees of freedom have been constrained, something which eliminates eigenvalues with null value without greatly impacting the overall results. These commands can be suppressed when clamped conditions are applied.

Eliminating boundary conditions is a simple case of commenting the lines where these commands are written to the text file and these can be found further down the file.
Lines controlling the output printing to the txt file with all Ansys commands. In this case only 1 line is constrained, dl1.

Txt_aux.m works in exactly the same way, the only difference being the command numbers. Boundary conditions should now be written in lines 163-169 and their suppression should be done in lines 267-273.

After inputting this information, all pre-processor related steps are completed and the program is ready to run.

B.2. Solving

The program’s master file is Initial.m and it is from here that all files are handled as well as the optimization.

Before doing so, the user is advised to look into the file fun.m and check the penalty scheme. The penalty scheme is defined at the start of this file and can be defined in any way desired. Nevertheless the rules described in chapter 5.3 should be respected to ensure the best results possible.
In *fun.m* file one also finds the line that controls the derivatives scaling. This is found at the end in line 149 and can be adjusted by changing the value inside `Resultante./(…)`.

Before running the file *Inicial.m* the user should decide whether or not the post-processor should be run. Post-processor is used for verifying the results and this is achieved by testing all possible combinations in *Ansys*, with the results being stored in the variable `Frequencias_Possiveis`. This step can be suppressed by commenting line 209 of *Inicial.m*.

To start optimizing problem all that remains is to run the file *Inicial.m*. The optimizer outputs some information about the eigenvalues at each iteration, as well as the computation time along with other data.

Information on the ten first frequencies of vibration in Hz can be found in variable `Frequencias_Hz`, the weight function’s values under `Pesos` and the design variables are stored in `x`. 
\\begin{verbatim}

**** FAIPAMAIZ.1.1 ****

FEASIBLE DIRECTION INTERIOR POINT ALGORITHM
THE CONSTRAINTS ARE NOT SCALED
THE CONSTRAINTS ARE NOT FILTERED
Quasi-Newton Algorithm
E approximates the inverse of the Hessian of the Lagrangian
INTERNAL LINEAR SYSTEMS:
Stopping Criteria for the CG:
   errcg1 = 0.001; errcg2 = 1e-006
THE BOX CONSTRAINTS ARE NOT REDUCED
Wolfe Criterion in the line search
The problem has no inequality constraints. Takes ifpoint=0.

Checking consistency of input data:

"Input data is consistent"


fun

\textit{nvar} = 16  \textit{ncon} = 0  \textit{nag} = 0  \textit{nbox} = 32

\textbf{ITER}  \textbf{OBJECTIVE}  \textbf{d0}  \textbf{lagran}  \textbf{errcg}  \textbf{STEP}  \textbf{nbind}  \textbf{iwood}
\hline
0 -7.24194e+006  \\
1 -8.48663e+006  0.261545  0.261545  0  1  0  2  \\
2 -1.12847e+007  7.07711  1.44568  0  0.111652  0  2  \\
3 -1.17018e+007  0.87558  1.06884  0  0.911666  0  2  \\
4 -1.19627e+007  0.245853  1.13615  0  1  0  2  \\
5 -1.20401e+007  0.122908  0.820165  0  1  0  2  \\
6 -1.20782e+007  0.0993856  0.475207  0  1  0  2  \\
7 -1.20825e+007  0.0595034  0.208251  0  1  0  2  \\
8 -1.20866e+007  0.0254714  0.045877  0  0.919275  0  2  \\
9 -1.20866e+007  0.0031535  0.00861877  0  0.00865714  0  2  \\
10 -1.20866e+007  0.00760627  0.00350422  0  0.0378388  0  2  \\
11 -1.20866e+007  0.000772543  0.000635037  0  2.23425e-006  0  2  \\
12 -1.20866e+007  0.00047095  0.00053132  0  0.798599  0  2  \\
13 -1.20866e+007  0.00219549  0.000707276  0  0.092433  0  2  \\
14 -1.20866e+007  0.000748043  0.000254227  0  0.00133076  0  2  \\
15 -1.20866e+007  0.00168199  0.000201839  0  0.000683989  0  2  \\
16 -1.20866e+007  0.000513337  1.5817e-005  0  0.221302  0  2  \\
17 -1.20866e+007  0.000176219  0.000212986  0  4.23676e-006  0  2  \\
18 -1.20866e+007  0.000612698  0.000212985  0  7.14764e-007  0  2  \\
\hline

STopping Criterion:

Reduction of the penalty function less than tolerance
Computation Time of the Optimization:
Elapsed time is 100.815635 seconds.
\end{verbatim}

Figure B-13 – Example of the feedback obtained from the optimizer.
B.3. Post-Processing

Post-processing is done at the file Verificacao.m. In it, all possible stacking orders are calculated and then analysed with Ansys. Since the number of possibilities increases exponentially, it should not be used in cases with many layers as it can take many hours to complete.

To use this capability the only input needed is in line 8, specifically, to the command `combinacoes = allcomb(...)`. Inside the brackets one should introduce the variable in `Orientacoes{2,2}` and then repeat the command a number of times equal to the number of layers.

An example of this command on a 5 layered case can be seen in Figure B-14.

```
6 - combinacoes = allcomb(Orientacoes{2,2},Orientacoes{2,2},Orientacoes{2,2},Orientacoes{2,2},Orientacoes{2,2});
```

*Figure B-14 – Total number of fibre possibilities in post-processing on a 5 layered case.*