

POLITECNICO DI TORINO



Faculty of Engineering
Master's Degree in Aerospace Engineering

Master Thesis

A hybrid global optimisation strategy for designing variable stiffness composites

Advisor:

prof. Erasmo Carrera

Co-advisors:

prof. Marco Montemurro

prof. Alfonso Pagani

Candidate:

Donati Francesca

March 2018

*"I would like to die on Mars.
Just not on impact."
Elon Musk*

*Al Cali,
mio nonno.*

Acknowledgements

This journey would not have been possible without the support of my family, professors and friends.

I would first like to thank my thesis advisor prof. Erasmo Carrera of the Polytechnic of Turin, for giving me the opportunity to take this challenging abroad experience, which taught me more than I could ever imagine.

Not less important I would like to express the deepest appreciation to my co-advisor at I2M Laboratory of the Institut de Mécanique et d'Ingénierie, prof. Marco Montemurro for his ongoing support; his office door was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work, but steered me in the right direction whenever he thought I needed it.

I would also like to express my gratitude to my co-advisor at Polytechnic of Turin, prof. Alfonso Pagani, for his patience and understanding, whose feedbacks about the development of this work have been proven to be invaluable.

My deepest appreciation goes to Mr. Giulio Costa, for having been greatly tolerant and supportive and whose support, encouragement, suggestions and comments were invaluable.

To my family, thank you for encouraging me in all of my pursuits and inspiring me to follow my dreams; I always knew that you all believed in me and wanted the best for me. I owe my deepest gratitude to my brother, Alessandro, for having been the best roommate ever, by supporting me every time I was in need and helping me through my hardest times.

I must thanks all the friends of the Polito Sailing Team, with whom I have worked over the last years for showing me what it means to be focused and dedicated to the carrying-out of an engineering team project.

A special thanks goes to the Gadz'Arts for making memorable my time in France.

Finally, I must express my very profound gratitude to my friends, old and new, for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Thank you.

Contents

Acknowledgements	II
1 Introduction	1
2 Optimization Theory	4
2.1 Fundamentals of Deterministic Optimization	4
2.2 Unconstrained and Constrained Optimization	4
2.3 Global Optimization and Convexity	5
2.4 Optimization Algorithms	5
3 Structural Materials Theory	7
3.1 Composite Materials	7
3.2 Mechanics of Composite Structures	8
3.3 Variable Stiffness Laminates	10
3.4 Polar Method	11
3.4.1 Polar formalism application on FSDT	13
3.5 Strain Energy Evaluation	16
3.5.0.1 Strain Energy Derivatives	17
4 Methodology	20
4.1 Unconstrained Methods	20
4.1.1 Line Search Approach	21
4.1.1.1 Step Length	22
4.1.1.2 Line Search Methods	23
4.1.2 Trust Region Approach	24
4.1.2.1 Subproblem Solution Methods	25
4.2 Constrained Methods	25
4.2.1 Linear Programming	26
4.2.2 Quadratic Programming	27
4.2.3 Sequential Quadratic Programming	27
4.3 B-spline Curves and Surfaces	28
4.4 Matlab <code>fmincon</code>	30
4.5 Genetic Algorithm BIANCA	31
4.6 ANSYS Mechanical APDL	32

4.6.1	Pre Integrated Matrices	33
5	Study cases	34
5.1	Description of the problem	34
5.2	Matemathical Statement of the Problem	35
5.3	Preliminary Analysis	36
5.4	Unidirectional Fiber Composite Plate	36
5.4.1	Simple Plate subjected to tensile stress	37
5.4.2	Simple Plate subjected to bending stress	38
5.5	Unidirectional Fiber Composite Hole Plate	39
5.5.1	Hole Plate subjected to tensile stress	40
5.5.2	Hole Plate subjected to bending stress	41
5.6	VAT laminate plate	43
5.6.1	BIANCA Environment	43
5.6.2	B-spline Domain	44
5.6.3	Plate subjected to a Tensile Force	46
5.6.3.1	First Configuration	46
5.6.3.2	Second Configuration	48
5.6.4	Plate subjected to a Bending Force	50
5.6.4.1	First Configuration	51
5.6.4.2	Second Configuration	53
6	Conclusions	57
6.1	Final Considerations	57
A	Algorithms	59
A.1	Unconstrained Algorithms	59
A.1.1	Line Search Algorithms	59
A.1.1.1	Step Length Algorithms	59
A.1.1.2	Descent Algorithms	61
A.1.2	Trust Region Algorithm	64
A.1.2.1	Trust Region Subproblem Algorithms	64
B	Homogenised Matrices and Derivatives	67
B.1	Homogenised Matrices	67
B.2	Homogenised Matrices Derivatives	69
B.2.1	Homogenised Matrix $[A^*]$ Derivatives	69
B.2.2	Homogenised Matrix $[D^*]$ Derivatives	69
B.2.3	Homogenised Matrix $[B^*]$ Derivatives	69
B.2.4	Homogenised Matrix $[H^*]$ Derivatives	69
	References	71

List of Figures

3.1	Classic Laminate Theory kinematic model	9
3.2	Laminate reference frame	10
3.3	Stiffness variation configuration	11
4.1	B-Spline Surface Control Net	29
4.2	B-Spline Surface	29
4.3	The architecture of the standard GA	31
4.4	Effect of the crossover operator on two homologous genes of the parents' couple.	32
4.5	Effect of the mutation operator on the bits of the single gene.	32
5.1	Unidirectional fiber composite plate BCs	37
5.2	Plate subjected to tensile stress model	37
5.3	Bending plate model	39
5.4	Unidirectional Fiber Composite Hole Plate BCs	40
5.5	Hole plate traction model	41
5.6	Hole plate bending model	42
5.7	Top half plate	45
5.8	Traction: first configuration. Optimal distribution of the anisotropic modulus R_{0K}^{A*} over the VAT plate resulting from the second step of the optimisation	47
5.9	Traction: first configuration. Optimal distribution of the anisotropic modulus R_1^{A*} over the VAT plate resulting from the second step of the optimisation	48
5.10	Traction: first configuration. Optimal distribution of the polar angle Φ_1^{A*} over the VAT plate resulting from the second step of the optimisation . . .	48
5.11	Traction: second configuration. Optimal distribution of the anisotropic modulus R_{0K}^{A*} over the VAT plate resulting from the second step of optimisation	50
5.12	Traction: second configuration. Optimal distribution of the anisotropic modulus R_1^{A*} over the VAT plate resulting from the first-step of optimisation	50
5.13	Traction: second configuration. Optimal distribution of the polar angle Φ_1^{A*} over the VAT plate resulting from the first-step of optimisation	51
5.14	Bending: first configuration. Optimal distribution of the anisotropic modulus R_{0K}^{A*} over the VAT plate resulting from the second step of the optimisation	52

5.15	Bending: first configuration. Optimal distribution of the anisotropic modulus R_1^{A*} over the VAT plate resulting from the second step of the optimisation	53
5.16	Bending: first configuration. Optimal distribution of the polar angle Φ_1^{A*} over the VAT plate resulting from the second step of the optimisation . . .	53
5.17	Bending: second configuration. Optimal distribution of the anisotropic modulus R_{0K}^{A*} over the VAT plate resulting from the second step of optimisation	55
5.18	Bending: second configuration. Optimal distribution of the anisotropic modulus R_1^{A*} over the VAT plate resulting from the first-step of optimisation	55
5.19	Bending: second configuration. Optimal distribution of the polar angle Φ_1^{A*} over the VAT plate resulting from the first-step of optimisation	56

List of Tables

5.1	Material properties of the carbon-epoxy pre-preg strip taken from [10]. . . .	34
5.2	Unidirectional Fiber Composite Plate Dimensions	36
5.3	BCs of the Unidirectional Fiber Composite Plate	36
5.4	Design space of the Unidirectional Fiber Composite Plate problem	37
5.5	MATLAB <code>fmincon</code> input and output parameters	38
5.6	Results for the unidirectional fiber composite plate traction analysis performed with ANSYS APDL and MATLAB <code>fmincon</code>	38
5.7	MATLAB <code>fmincon</code> input and output parameters	38
5.8	Results for the unidirectional fiber composite plate traction analysis performed with ANSYS APDL and MATLAB <code>fmincon</code>	39
5.9	Unidirectional Fiber Composite Hole Plate Dimensions	40
5.10	BCs of the Unidirectional Fiber Composite Hole Plate	40
5.11	MATLAB <code>fmincon</code> input and output parameters	40
5.12	Results for the unidirectional fiber composite hole plate traction analysis performed with ANSYS APDL and MATLAB <code>fmincon</code>	41
5.13	MATLAB <code>fmincon</code> input and output parameters	41
5.14	Results for the unidirectional fiber composite hole plate bending analysis performed with ANSYS APDL and MATLAB <code>fmincon</code>	42
5.15	Input parameters for <code>.gen</code> file GA BIANCA	44
5.16	Traction: first configuration. Optimum value of R_{0K}^{A*} for each control point of the B-spline surface after <code>fmincon</code> optimisation	46
5.17	Traction: first configuration. Optimum value of R_1^{A*} for each control point of the B-spline surface after <code>fmincon</code> optimisation	47
5.18	Traction: first configuration. Optimum value of Φ_1^{A*} for each control point of the B-spline surface after <code>fmincon</code> optimisation	47
5.19	Traction: second configuration. Optimum value of R_{0K}^{A*} for each control point of the B-spline surface after <code>fmincon</code> optimisation	49
5.20	Traction: second configuration. Optimum value of R_1^{A*} for each control point of the B-spline surface after <code>fmincon</code> optimisation	49
5.21	Traction: second configuration. Optimum value of Φ_1^{A*} for each control point of the B-spline surface after <code>fmincon</code> optimisation	49
5.22	Bending: first configuration. Optimum value of R_{0K}^{A*} for each control point of the B-spline surface after <code>fmincon</code> optimisation	51

5.23	Bending: first configuration. Optimum value of R_1^{A*} for each control point of the B-spline surface after fmincon optimisation	52
5.24	Bending: first configuration. Optimum value of Φ_1^{A*} for each control point of the B-spline surface after fmincon optimisation	52
5.25	Bending: second configuration. Optimum value of R_{0K}^{A*} for each control point of the B-spline surface after fmincon optimisation	54
5.26	Bending: second configuration. Optimum value of R_1^{A*} for each control point of the B-spline surface after fmincon optimisation	54
5.27	Bending: second configuration. Optimum value of Φ_1^{A*} for each control point of the B-spline surface after fmincon optimisation	54
5.28	Summary of the strain energy values	55

Chapter 1

Introduction

The present work takes origin from studies carried out at the Institut de Mécanique et d'Ingénierie (I2M) Laboratory of Bordeaux, France, about the study and research of an hybrid global optimization strategy for designing variable angle tow (VAT) laminates.

In the aerospace and aeronautical engineering, the main topics which must be considered in the structural study field are essentially three: *structure*, defined as an aggregation of different components which aim to sustain loads, *composite materials*, stated as a material composed of at least two different materials, combined in such way to get properties superior to those of the individual constituents and *optimisation* defined as a systematic process which allows the designer to find an optimum configuration of the composite structure at each *pertinent scale*, according to the requirement provided in the specifications for the problem at hand. Starting from these definitions, *structural optimisation* may be defined as "the rational establishment of structural design that is the best of all possible design within a prescribed objective and given set of geometrical and/or behavioral limitations"[3].

Generally speaking, composite structures should be designed to withstand applied forces without exceeding acceptable values of deformations, as well as being as cheaper and lighter as possible.

In this background VAT represent a generalisation of the concept of classical straight fibre-reinforced composites. The orientation angle of each constitutive ply can be locally arranged in order to optimise the behaviour of the composite. In particular, for each layer the fibre path varies pointwise and can be adapted to the stress field in order to maximise the stiffness by minimising, simultaneously, the mass of the components. Of course, VAT composites can be tailored only through modern manufacturing processes like the automated fibre placement (AFP) technique. Although VAT composites show superior performances when compared to classical straight-fibre format, on the other hand the design process is quite cumbersome and requires the implementation of an *ad-hoc* multi-scale optimisation strategy able to optimise locally the VAT composite at each pertinent scale (i.e., typically mesoscopic and macroscopic ones). Of course the problem of designing a VAT composite is usually characterised by a huge number of design variables when compared to classical composites. Starting from research studies carried out at I2M laboratory

on this subject [10] and [9], this thesis aims at proposing a hybrid analytical/numerical approach for determining the derivatives of some relevant quantities (e.g. stiffness, feasibility constraints, etc.) for VAT composites. This approach will be then integrated in the framework of the multi-scale two-level (MS2L) optimisation strategy developed at I2M [10] and [9].

In the framework of the MS2L methodology, the design problem is split into two sub-problems. At the first-level of the strategy (macroscopic scale) the goal is to determine the optimum distribution of the laminate stiffness properties over the structure, while the second step aims at retrieving the optimum fibres-path in each layer meeting all the requirements provided by the problem at hand. This thesis will focus only on the first-level of the strategy.

The MS2L optimisation strategy relies on: (a) the polar formalism for describing the behaviour of the VAT laminate, (b) the B-Spline hypersurfaces formalism for describing the spatial variation of the laminate stiffness and (c) an hybrid optimisation tool (genetic and gradient-based algorithms) to perform the solution search.

The *polar formalism*, firstly introduced by Verchery in [4], is an algebraic technique to represent a plane tensor using only tensor invariants. The advantage of this framework is that it is based on true tensor invariants having also a geometrical meaning which can be used as intrinsic quantities characterizing the material as well as the related elastic symmetries, allowing to study their effect in the process of finding the optimal solution [1].

The *B-spline hypersurface* framework allows reducing the number of design variables required to describe the pointwise variation of the laminate polar parameters (and hence the laminate stiffness matrices) by ensuring, simultaneously, the continuity of the fibres path over the structure (through the continuity of the stiffness properties at the macroscopic scale).

The software and tools used to perform the analysis are essentially three: the Genetic Algorithm BIANCA [7], the ANSYS automatic parametric design language (APDL) is utilised to build the parametric finite element (FE) model of the VAT composite which is interfaced with the previous algorithm during the whole optimisation process.

The originality of this work is twofold. The first original point is the description of the macroscopic behaviour of the laminate through the laminate polar parameters. The second one is the use determination of the derivative of some peculiar properties of the VAT composite, i.e. the energy and the feasibility constraints in the framework of the B-Spline hypersurfaces formalism, in order to perform the gradient descent optimisation.

The rest of the manuscript is structured as follows: second and third chapters present the key points of mathematical optimisation theory and the composite materials general theory, by focusing especially on the notation of the polar method and the energy evaluation. The fourth chapter introduces the proposed optimisation strategy to perform the solution search for different kind of problems. Firstly, the proposed approach is applied to the minimisation of the strain energy of a classical composite composed of unidirectional plies (straight fibre format) by considering feasibility constraints on the laminate polar parameters as well as a geometric constraint on the overall number of plies. In a second time, the same problem has been solved by considering a VAT composite. For each problem two

different geometric configurations of the structure have been considered (a simple rectangular plate and a plate with an hole) subject to different boundary conditions (BCs). The numerical results of the approach are presented and discussed in chapter five. Finally, some conclusive remarks and future perspectives ends the manuscript.

The present work do not pretend to be exhaustive since the study fields taken into account are huge and high skilled. However, the reader interested is addressed to the appropriate references listed in the bibliography for a deeper insight in the matter.

As a conclusive remark, a short note: in the following, the terminology, nomenclature and mathematical formulation arising from various engineering and mathematical fields are considered standard and known by the reader, although quick references are made when necessary to facilitate the understanding of particular topics.

Chapter 2

Optimization Theory

2.1 Fundamentals of Deterministic Optimization

Mathematical Optimization or Mathematical Programming, alternatively spelled *optimization* is the selection of a best element (with regard to some criterion) from some set of available alternatives. In the simplest case, an optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function; more generally, optimization includes finding "best available" values of some objective function given a defined domain (or input), including a variety of different types of objective functions and different types of domains. We use the following notation:

- x is the vector of *variables*, also called *unknowns* or *parameters*;
- f is the *objective function*, a (scalar) function of x that we want to maximize or minimize;
- c_i are *constraint functions*, which are scalar functions of x that define certain equations and inequalities that the unknown vector x must satisfy.

Using this notation, the optimization problem can be written as follows:

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} c_i(x) = 0 & i \in \mathcal{E} \\ c_i(x) \leq 0 & i \in \mathcal{I} \end{cases} \quad (2.1)$$

where $\mathcal{E}$ and $\mathcal{I}$ are sets of indices for equality and inequality constraints.

2.2 Unconstrained and Constrained Optimization

Quoting [17], problems with the general form (2.1) can be classified according to the nature of the objective function and constraints (linear, nonlinear, convex), the number of variables (large or small), the smoothness of the functions (differentiable or nondifferentiable), and so on. An important distinction is made between problems that have constraints on

the variables and those that do not.

Unconstrained optimization problems, for which we have $\mathcal{I}=\mathcal{E}=0$ in (2.1), are often used as well as reformulations of constrained optimization problems, in which the constraints are replaced by penalization terms added to objective function in order to avoid constraint violations.

Constrained optimization problems occur from models in which constraints are present; these constraints may be a simple bound, a general linear or a nonlinear constraints. When the objective function and all the constraints are linear functions of x and the problem is defined as a *Linear programming* problem while, when at least some of the constraints or the objective are nonlinear functions, the problems is defined as a *Nonlinear programming* problem.

2.3 Global Optimization and Convexity

With regard to nonlinear optimization problems, it has to be said that many algorithms seek only a local solution, i.e. a point at which the objective function is smaller than at all other feasible nearby points, and don't always find the global solutions requested in some applications, due to the difficulty in recognising and locating them. Convex minimization is a subfield of optimization that studies the problem of minimizing convex functions over convex sets. The convexity makes optimization easier than the general case since local minimum must be a global minimum, and first-order conditions are sufficient conditions for optimality. If the objective function in the optimization problem (2.1) and the feasible region are both convex, then any local solution of the problem is in fact a global solution.

2.4 Optimization Algorithms

Optimization algorithms are iterative. They begin with an initial guess of the variable x and generate a sequence of approximatives calculations (called “iterates”) until they find a solution. The algorithms can be distinguishes one from another depending on the strategy used to evaluate the new iterate: some algorithms make use of the values of the objective function f , the constraint functions c_i , and even the first and second derivatives of both these functions; others use informations from previous iterations while some use the local informations at the current point. Regardless of these different approaches, all good optimization algorithms should be:

- *Robust*, i.e. the algorithm should perform well on a wide variety of problems for all acceptable starting points;
- *Efficient*, i.e. the algorithm shouldn't require excessive time or storage to be performed;
- *Accurate*, i.e. the algorithm should be able to identify a solution with precision, without being too much sensitive to errors in the data or to errors due to computer implementation.

These goals may conflict: a rapidly convergent method for a large unconstrained nonlinear problem may require too much computer storage while a robust method may also be the slowest. According to [17], tradeoffs between convergence rate and storage requirements, and between robustness and speed, and so on, are central issues in numerical optimization.

Chapter 3

Structural Materials Theory

3.1 Composite Materials

A *composite material* consists in two or more materials, combined on a macroscopic scale in order to obtain a third material which shows the best qualities of its constituents and sometimes even qualities that the constituents don't possess. Commonly there are four classes of composite materials that are actually used:

- *Fibrous composite materials* made up of fibers and matrix,
- *Particulate composite materials* made up of macro size particles in a matrix,
- *Laminated composite materials* made up of layers of various materials, including composites of the first two types,
- Combination of the first three categories.

Speaking about the first category, *fibers* are the reinforcement material, where the desired properties are maximized in a given direction, while *matrix* is the binder material which duty is to support and protect the fibers, along to transfer the stresses between broken fibers. Typical fibers for composite applications have high strength and stiffness in fiber direction, while they are weak and flexible perpendicular to it; also other physical properties, like electrical or thermal conductivity may be totally different when measured along or perpendicular to a fiber. Fibers are stiffer and stronger than the same material in bulk form and have a very high length-to-diameter ratio. Instead, the matrix is considerably lower in density, stiffness and strength and usually have its bulk-form properties. In the case of structural applications, fiber-reinforced composite materials are usually a thin layer called *lamina* or *ply* that represents the fundamental building block of the composite and consists of many fibers embedded in a matrix material. A collection of laminae stacked to obtain the desired stiffness and thickness is called *laminate* and the sequence of the various laminae's orientations is called *lamination scheme* or *stacking sequence*; usually the layers are bounded together with the same matrix material used in the single lamina. The lamination scheme and material properties of individual lamina provide an added

flexibility to designers to tailor the stiffness and strength of the laminate to match the structural stiffness and strength requirements. The main disadvantages of laminates made of fiber-reinforced composite materials are delamination and fiber debonding: *delamination* is due to the mismatch of material properties between layers, which produces shear stresses between the layers, while *fiber debonding* is caused by the mismatch of material properties between matrix and fiber. An other drawback presents in this kind of laminates is that, during the manufacturing, material defects such as interlaminar voids, incorrect orientation or damaged fibers may be introduced affecting the performances of the total laminate.

3.2 Mechanics of Composite Structures

The design of a laminate is a complex and difficult issue; while for a homogeneous material plate, the only design variable is the thickness, the laminate has final mechanical characteristics that depend by the ones of the lamina, the numbers of the laminae and their relative orientation. The design phase include the design of both resistance and stiffness along the elastic response. The Classic Laminate Theory (CLT) aims to provide a mathematical model able to synthesize the elastic response of a laminate like it was made up by only one equivalent layer. The hypothesis behind the classic laminate theory are:

- The laminate consists of perfectly bonded layers. There is no slip between the adjacent layers. In other words, it is equivalent to saying that the displacement components are continuous through the thickness.
- Each lamina is considered to be a homogeneous layer such that its effective properties are known.
- Each lamina is in a state of plane stress.
- The individual lamina can be isotropic, orthotropic or transversely isotropic.
- The laminate deforms according to the Kirchhoff-Love assumptions for bending and stretching of thin plates (as assumed in classical plate theory). The assumptions are:
 - The normals to the mid-plane remain straight and normal to the midplane even after deformation.
 - The normals to the mid-plane do not change their lengths.

The choice of this kinematic model results in the presence of mechanical consequences, like the incapability to tracking down shear deformations on the plate thickness. Considering Figure 3.1, the displacement field of a point P of coordinates (x, y, z) can be written as follow:

$$\mathbf{u} = \begin{Bmatrix} u_0(x, y) - z \frac{\partial w_0(x, y)}{\partial x} \\ v_0(x, y) - z \frac{\partial w_0(x, y)}{\partial y} \\ w_0(x, y) \end{Bmatrix} \quad (3.1)$$

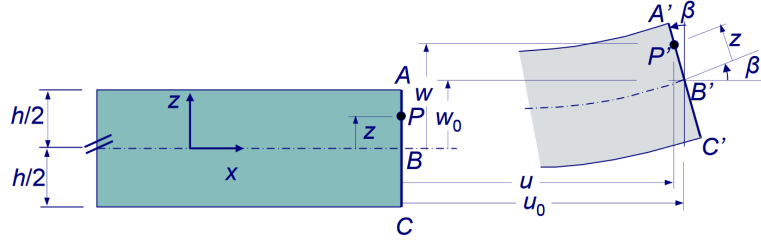


Figure 3.1: Classic Laminate Theory kinematic model

from which it can be seen that the displacement field is linear in z . From the assumptions on the kinematic behaviour, it follows that the strain field is an *in-plane deformation field* and the middle plane small strain tensor $\boldsymbol{\varepsilon}^0$ can be computed as

$$\boldsymbol{\varepsilon}^0 = \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \varepsilon_s^0 \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u_0(x,y)}{\partial x} \\ \frac{\partial v_0(x,y)}{\partial y} \\ \frac{\partial u_0(x,y)}{\partial y} + \frac{\partial v_0(x,y)}{\partial x} \end{Bmatrix} \quad (3.2)$$

while the opposite of the middle plane curvature tensor $\boldsymbol{\kappa}$, thanks to small displacements and rotations hypothesis, can be computed as

$$\boldsymbol{\kappa} = \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_s \end{Bmatrix} = - \begin{Bmatrix} \frac{\partial^2 w_0(x,y)}{\partial x^2} \\ \frac{\partial^2 w_0(x,y)}{\partial y^2} \\ 2 \frac{\partial^2 w_0(x,y)}{\partial y \partial x} \end{Bmatrix} \quad (3.3)$$

Finally the strain tensor can be written as

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^0 + z\boldsymbol{\kappa} \rightarrow \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_s \end{Bmatrix} = \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \varepsilon_s^0 \end{Bmatrix} + z \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_s \end{Bmatrix} \quad (3.4)$$

The knowledge of the strain field allows to compute the stress field and, consequently, the internal actions resultants. The stress field is generally considered plane, like the strain field, even if it's not, due to the small values of stress σ'_3 compared to the eligible ones and to the small thickness of the plate. The link between σ and ε is given by the Hooke's Law:

$$\{\sigma\} = [Q] \{\varepsilon\} \quad (3.5)$$

where $[Q]$ is the reduced stiffness matrix.

The resultants calculated are only the membrane and bending ones, since no stresses are acting in the z -direction, and are computed, through Voigt's notation, as follows:

$$\begin{Bmatrix} N_x \\ N_y \\ N_s \end{Bmatrix} = \begin{Bmatrix} \int_{-h/2}^{h/2} \sigma_x dz \\ \int_{-h/2}^{h/2} \sigma_y dz \\ \int_{-h/2}^{h/2} \sigma_s dz \end{Bmatrix}, \begin{Bmatrix} M_x \\ M_y \\ M_s \end{Bmatrix} = \begin{Bmatrix} \int_{-h/2}^{h/2} \sigma_x z dz \\ \int_{-h/2}^{h/2} \sigma_y z dz \\ \int_{-h/2}^{h/2} \sigma_s z dz \end{Bmatrix} \quad (3.6)$$

Since the strain field is unique for all the layers and described by two tensor that don't depend on the z coordinate, referring to Figure 3.2 and re-writing equation (3.6) in an extended form, it's possible to establish the following relations:

$$\begin{aligned} [A] &= \sum_{k=1}^n \int_{z_{k-1}}^{z_k} [Q_k] (\delta_k) dz \\ [B] &= \sum_{k=1}^n \int_{z_{k-1}}^{z_k} [Q_k] (\delta_k) z dz \\ [D] &= \sum_{k=1}^n \int_{z_{k-1}}^{z_k} [Q_k] (\delta_k) z^2 dz \end{aligned} \quad (3.7)$$

that allows to obtain the *fundamental law of laminates* which matrix format is given below

$$\begin{Bmatrix} \{N\} \\ \{M\} \end{Bmatrix} = \begin{bmatrix} [A] & [B] \\ [B] & [D] \end{bmatrix} \begin{Bmatrix} \{\varepsilon^0\} \\ \{\kappa\} \end{Bmatrix} \quad (3.8)$$

and represents the real founding law under the assumptions made.

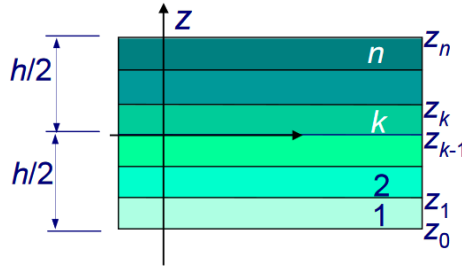


Figure 3.2: Laminite reference frame

3.3 Variable Stiffness Laminates

In the traditional design of the composite laminates previously introduced, the purpose is to find out the material arrangement that best satisfies some requirements like strength, stiffness and cost; fiber orientation angles are constant for each layer of the laminate and usually limited to 0, 90, and ± 45 degrees. This approach ignored the full potential of

composites because only a limited number of possible combinations of fiber orientation and stacking sequence can be used. Variable stiffness laminates are laminates within which stiffness properties are a function of spatial location; this means, in other words, that stiffness properties change point to point. Through tailoring of fiber orientations and laminate thickness spatially in an optimal fashion, mechanical properties of a part can be improved. This stiffness variation may be *discrete*, by defining several different patches within a laminate, or *continuous*, by varying the fiber angle orientation continuously within a ply's domain.

The composite laminate showing a variable angle tow are called VAT composite.

The variation of fiber orientation is achieved by Automated Fiber Placement (AFP) machines which allow to place the fibre, i.e. the tow, along a curvilinear path. The utilisation

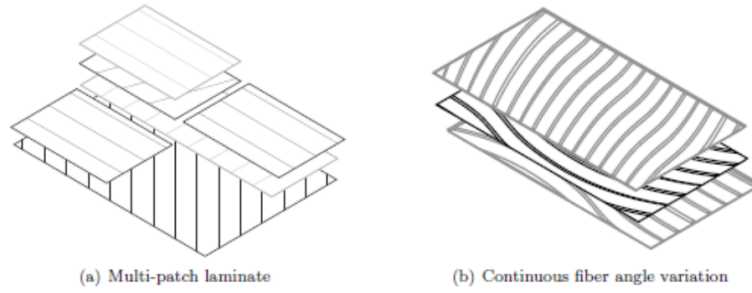


Figure 3.3: Stiffness variation configuration

of VAT laminates increases considerably the complexity of the design process, due to the increase of the design variables number but allows the designer to obtain different and non-conventional solutions saving weight, improving mechanical properties and performances.

3.4 Polar Method

Anisotropy is the dependence of a physical property on the direction [13]. The intrinsic difficulty of anisotropy is that the response depends on a considerable number of parameters and that these parameters depend upon the reference frame chosen to describe the phenomenon. This means that anisotropy is often described by non invariants quantities which requires us to specify the reference frame along with the parameters that represent the behaviour of the material only in that specific reference frame. Apart this difficulty, it has to take into account the representation of the tensor representation; normally a Cartesian representation is used which still presents the drawbacks listed above: frame-dependance and no-intrinsic representation. Therefore it seems natural to use, in some cases, a tensor invariants representation in order to describe the behaviour of a continuum by material intrinsic quantities. The polar method is a mathematical technique that allows to describe the n -rank tensor introduced above with a set of tensor invariants. A second-rank symmetric tensor T_{ij} , ($i, j = 1, 2$), within the local frame of the elementary

ply, can be defined as follow:

$$\begin{aligned} T_{11} &= T + R\cos 2\Phi, \\ T_{12} &= R\sin 2\Phi, \\ T_{22} &= T - R\cos 2\Phi, \end{aligned} \quad (3.9)$$

where T , R and Φ are respectively the isotropic modulus, the deviatoric modulus and the polar angle. In the case of a second-rank plane symmetric tensor, we can arbitrarily choose Φ in order to fix the reference frame, while the two polar moduli T and R are tensor invariants.

The relations between the polar moduli and the component of tensor are:

$$\begin{aligned} T &= \frac{T_{11} + T_{22}}{2}, \\ Re^{i2\Phi} &= \frac{T_{11} - T_{22}}{2} + iT_{12}. \end{aligned} \quad (3.10)$$

In case of a frame rotation of θ amplitude, the Cartesian components (3.9) can be expressed as follows:

$$\begin{aligned} T_{xx} &= T + R\cos 2(\Phi - \theta), \\ T_{xy} &= R\sin 2(\Phi - \theta), \\ T_{yy} &= T - R\cos 2(\Phi - \theta). \end{aligned} \quad (3.11)$$

In the case of a fourth-rank elasticity tensor, Verchery introduced the polar components T_0 , T_1 , R_0 , R_1 , Φ_0 and Φ_1 , where T_0 and T_1 are the isotropic moduli, R_0 and R_1 are the anisotropic ones, while Φ_0 and Φ_1 are the polar angles [4]. As stated for the second-rank plane tensor, even for the fourth-rank elasticity tensor the tensor invariants are T_0 , T_1 , R_0 , R_1 together with the difference $\Phi_0 - \Phi_1$, which means that one of the two polar angles can be arbitrarily chosen in order to fix the reference frame. These entities can be used to render explicit the tensor Cartesian components in a polar representation as:

$$\begin{aligned} T_{1111} &= T_0 + 2T_1 + R_0\cos(4\Phi_0) + 4R_1\cos(2\Phi_1), \\ T_{1122} &= -T_0 + 2T_1 - R_0\cos(4\Phi_0), \\ T_{1112} &= R_0\sin(4\Phi_0) + 2R_1\sin(2\Phi_1), \\ T_{2222} &= T_0 + 2T_1 + R_0\cos(4\Phi_0) - 4R_1\cos(2\Phi_1), \\ T_{2212} &= -R_0\sin(4\Phi_0) + 2R_1\sin(2\Phi_1), \\ T_{1212} &= T_0 - R_0\cos(4\Phi_0). \end{aligned} \quad (3.12)$$

The inverse relations can be written as follows:

$$\begin{aligned} 8T_0 &= T_{1111} - 2T_{1122} + 4T_{1212} + T_{2222}, \\ 8T_1 &= T_{1111} + 2T_{1122} + T_{2222}, \\ 8R_0e^{i4\Phi_0} &= T_{1111} - 2T_{1122} - 4T_{1212} + T_{2222} + 4i(T_{1112} - T_{2212}), \\ 8R_1e^{i2\Phi_1} &= T_{1111} - T_{2222} + 2i(T_{1112} + T_{2212}). \end{aligned} \quad (3.13)$$

Again, in case of a frame rotation of θ amplitude, the Cartesian components of the fourth-rank tensor are:

$$\begin{aligned}
 T_{xxxx} &= T_0 + 2T_1 + R_0 \cos(4\Phi_0 - \theta) + 4R_1 \cos(2\Phi_1 - \theta), \\
 T_{xxyy} &= -T_0 + 2T_1 - R_0 \cos(4\Phi_0 - \theta), \\
 T_{xxxy} &= R_0 \sin(4\Phi_0 - \theta) + 2R_1 \sin(2\Phi_1 - \theta), \\
 T_{yyyy} &= T_0 + 2T_1 + R_0 \cos(4\Phi_0 - \theta) - 4R_1 \cos(2\Phi_1 - \theta), \\
 T_{yyxy} &= -R_0 \sin(4\Phi_0 - \theta) + 2R_1 \sin(2\Phi_1 - \theta), \\
 T_{xyxy} &= T_0 - R_0 \cos(4\Phi_0 - \theta).
 \end{aligned} \tag{3.14}$$

The polar method is particularly important when a fourth-rank elasticity plane tensor is taken into account because, in this specific case, the polar invariants are linked to the symmetries of the tensor and, consequently, they immediately acquire physical meaning. The *algebraic* characterization of the (elastic) symmetries, offered by the polar method, can be seen as a different solution technique, instead of the standard geometrical approach, to the problem of finding the elastic symmetries. In the study of a fourth-rank elasticity plane tensor four types of elastic symmetries exist:

$$\text{Ordinary orthotropy} \quad \Phi_0 - \Phi_1 = K \frac{\pi}{4}, \quad K = 0, 1 \tag{3.15}$$

$$R_0 - \text{Orthotropy} \quad R_0 = 0 \tag{3.16}$$

$$\text{Square symmetry} \quad R_1 = 0 \tag{3.17}$$

$$\text{Isotropy} \quad R_0 = R_1 = 0. \tag{3.18}$$

The studied cases presented in this work will take into account only the first symmetry (3.15).

For more details the reader is addressed to [6] and [13].

3.4.1 Polar formalism application on FSDT

In order to simplify the following section, in the writing of all the governing equations of the laminate, the Voigt's notation will be used. The passage from the previously notation can be easily denoted by the following relationships:

$$\begin{aligned}
 \{11, 22, 33, 32, 31, 21\} &\Longleftrightarrow \{1, 2, 3, 4, 5, 6\}, \\
 \{xx, yy, zz, zy, zx, yx\} &\Longleftrightarrow \{x, y, z, q, r, s\}.
 \end{aligned} \tag{3.19}$$

The structure considered is a multilayer plate constituted by a n number of identical layers; this configuration implies that all the layers have same thickness and material properties. Let us define δ_k the fiber orientation angle of the k -th ply, t_{ply} the thickness of the single layer and $h = nt_{ply}$ the total thickness. Accordingly to [12], the constitutive law of the laminated plate are:

$$\begin{Bmatrix} \{N\} \\ \{M\} \end{Bmatrix} = \begin{bmatrix} [A] & [B] \\ [B] & [D] \end{bmatrix} \begin{Bmatrix} \{\varepsilon_0\} \\ \{\kappa\} \end{Bmatrix} \tag{3.20}$$

$$\{F\} = [H] \{\gamma_0\} \quad (3.21)$$

where $[A]$ is the membrane stiffness matrix, $[B]$ is the membrane/bending coupling stiffness matrix, $[D]$ is the bending stiffness matrix and $[H]$ is the out-of-plane shear stiffness matrix. Furthermore, $\{N\}$, $\{M\}$ and $\{F\}$ are vectors containing respectively the membrane forces, the bending moments and the shear forces per unit length, while $\{\varepsilon_0\}$, $\{\chi_0\}$ and $\{\gamma_0\}$ are the vectors corresponding to the in-plane strains, the curvatures and the out-of-plane shear strains evaluated in the middle plane of the plate. The in-plane stiffness matrices introduced before can be written in the following form:

$$\begin{aligned} [A] &= \frac{h}{n} \sum_{k=1}^n [Q(\delta_k)], \\ [B] &= \frac{1}{2} \left(\frac{h}{n}\right)^2 \sum_{k=1}^n b_k [Q(\delta_k)], \\ [D] &= \frac{1}{12} \left(\frac{h}{n}\right)^3 \sum_{k=1}^n d_k [Q(\delta_k)], \end{aligned} \quad (3.22)$$

where $[Q(\delta_k)]$ is the in-plane reduced stiffness matrix of the single ply. The out-of-plane shear stiffness matrix $[H]$ can be represented as follows:

$$[H] = \begin{cases} \frac{h}{n} \sum_{k=1}^n [\hat{Q}(\delta_k)] & \text{(basic),} \\ \frac{5h}{12n^3} \sum_{k=1}^n (3n^2 - \delta_k) [\hat{Q}(\delta_k)] & \text{(modified).} \end{cases} \quad (3.23)$$

where $[\hat{Q}(\delta_k)]$ is the out-of-plane shear stiffness matrix of the single ply. The studied cases presented in this work will take into account the basic expression of (3.23).

The components of the matrices $[Q(\delta_k)]$ and $[\hat{Q}(\delta_k)]$ can be expressed by using the polar formalism in the following way:

$$\begin{aligned} Q_{xx} &= T_0 + 2T_1 + R_0 \cos(4\Phi_0 + \delta_k) + 4R_1 \cos(2\Phi_1 + \delta_k), \\ Q_{xy} &= -T_0 + 2T_1 - R_0 \cos(4\Phi_0 + \delta_k), \\ Q_{xs} &= R_0 \sin(4\Phi_0 + \delta_k) + 2R_1 \sin(2\Phi_1 + \delta_k), \\ Q_{yy} &= T_0 + 2T_1 + R_0 \cos(4\Phi_0 + \delta_k) - 4R_1 \cos(2\Phi_1 + \delta_k), \\ Q_{ys} &= -R_0 \sin(4\Phi_0 + \delta_k) + 2R_1 \sin(2\Phi_1 + \delta_k), \\ Q_{ss} &= T_0 - R_0 \cos(4\Phi_0 + \delta_k). \end{aligned} \quad (3.24)$$

and

$$\begin{aligned} \hat{Q}_{qq} &= T + R \cos 2(\Phi + \delta_k), \\ \hat{Q}_{qr} &= R \sin 2(\Phi + \delta_k), \\ \hat{Q}_{rr} &= T - R \cos 2(\Phi + \delta_k). \end{aligned} \quad (3.25)$$

Regarding this work, as previously said, the symmetry 3.15 is taken into account, with K value settled as 0. The value of K is very important in optimization problems: in fact,

it has been seen in several cases that changing K from 0 to 1 or vice-versa transforms an optimal solution into an anti-optimal one. Referring to Eq. 3.15 symmetry, in order to express every elements of the elastic tensor, i.e. of the stiffness matrix, as a function of R_0 , R_1 and Φ_1 , and re-writing as $\Phi_0 = \Phi_1 + K\frac{\pi}{4}$, the following expressions must be introduced:

$$\begin{aligned} \cos(4\Phi_0) &= \cos(4\Phi_1 + K\pi) = \cos(4\Phi_1), \\ \sin(4\Phi_0) &= \sin(4\Phi_1 + K\pi) = \sin(4\Phi_1), \\ \cos(2\Phi_0) &= \cos(2\Phi_1 + K\frac{\pi}{2}) = \cos(2\Phi_1), \\ \sin(2\Phi_0) &= \sin(2\Phi_1 + K\frac{\pi}{2}) = \sin(2\Phi_1). \end{aligned} \quad (3.26)$$

In this light, Eq. 3.24 can be re-written as:

$$\begin{aligned} Q_{xx} &= T_0 + 2T_1 + R_0K\cos(4\Phi_1) + 4R_1\cos(2\Phi_1), \\ Q_{xy} &= -T_0 + 2T_1 - R_0K\cos(4\Phi_1), \\ Q_{xs} &= R_0K\sin(4\Phi_1) + 2R_1\sin(2\Phi_1), \\ Q_{yy} &= T_0 + 2T_1 + R_0K\cos(4\Phi_1) - 4R_1\cos(2\Phi_1), \\ Q_{ys} &= -R_0K\sin(4\Phi_1) + 2R_1\sin(2\Phi_1), \\ Q_{ss} &= T_0 - R_0K\cos(4\Phi_1). \end{aligned} \quad (3.27)$$

For the sake of clarity and in order to simplify and better understanding the mechanical response of the structure, the following homogenised matrices will be taken into account:

$$\begin{aligned} [A^*] &= \frac{1}{h} [A], \\ [B^*] &= \frac{2}{h^2} [B], \\ [D^*] &= \frac{12}{h^3} [D], \\ [H^*] &= \begin{cases} \frac{1}{h} [H] & \text{(basic),} \\ \frac{12}{5h} [H] & \text{(modified).} \end{cases} \end{aligned} \quad (3.28)$$

These homogenised matrices can be expressed by using the polar parameters which the extended form can be found in (B.1). In conclusion, since in the majority of the real engineering applications the laminate is considered uncoupled, when a *quasi-homogeneous* laminate is used the following properties are satisfied:

$$\begin{aligned} [B^*] &= [0], \\ [C^*] &= [A^*] - [D^*] = [0], \end{aligned} \quad (3.29)$$

where $[C^*]$ is the homogeneity matrix. From this it can be seen that the only quantities used are the anisotropic polar moduli and the polar angles of the membrane stiffness matrix and that this generally applies even in the framework of the FSDT; moreover the matrix $[D^*]$ is assumed to be the same as $[A^*]$.

3.5 Strain Energy Evaluation

The concept of strain energy is of fundamental importance in applied mechanics and is defined as the energy stored in a body due to deformation caused when external loads are applied. From the Virtual Displacement Principle (VDP) definition, claiming that *external virtual work is equal to internal virtual work when equilibrated forces and stresses undergo unrelated but consistent displacements and strains*, in the static case, we know that:

$$\delta L_i = \delta L_e \quad (3.30)$$

where δL_i represents the internal virtual work and δL_e is the external virtual work. The matrix form of (3.30) is:

$$\delta L_i = \int_V \{\sigma\}^T \{\delta\varepsilon\} dV \quad (3.31)$$

By replacing (3.5) in (3.31) we can write:

$$\delta L_i = \int_V \{\varepsilon\}^T [Q] \{\delta\varepsilon\} dV \quad (3.32)$$

since $[Q] = [Q]^T$. From the cinematic equation we can write the following relation:

$$\{\varepsilon\} = [b] \{u\} \quad (3.33)$$

where $\{u\}$ is the displacements vector and $[b]$ is a differential operator. Equation (3.31) can be re-written as follows

$$\delta L_i = \int_V \left\{ \{u\}^T [b]^T \right\} [Q] \{\delta([b] \{u\})\} dV \quad (3.34)$$

Introducing $[N]$ as the shape function matrix, the displacement vector $\{u\}$ can be linked with the nodal displacement vector using

$$\{u\} = [N] \{U\}. \quad (3.35)$$

It's now obviously convenient to express $\{\delta\varepsilon\}$ as a function of $\{U\}$ as

$$\{\delta\varepsilon\} = [b] [N] \{U\} \quad (3.36)$$

and introducing the deformations interpolation matrix $[B] = [b] [N]$ we obtain

$$\delta L_i = \int_V \left\{ \{U\}^T [B]^T \right\} [Q] \{\delta([B] \{U\})\} dV \quad (3.37)$$

where the nodal displacement vector $\{U\}$ is independent of interpolation. This allows to re-written (3.31) as

$$\delta L_i = \{U\}^T \int_V [B]^T [Q] [B] dV \{\delta U\} \quad (3.38)$$

and in a compact form

$$\delta L_i = [U]^T [K] \{\delta U\} \quad (3.39)$$

where

$$[K] = \int_V [B]^T [Q] [B] dV \quad (3.40)$$

is the **generic expression of the element stiffness matrix**.

At this point it's easy to written the strain energy expression form the VDP as follows:

$$\mathcal{E} = \frac{1}{2} \int_V \{\sigma\}^T \{\varepsilon\} dV \quad (3.41)$$

In the FSDT field, we need to add the shear contribute to the fundamental laws (3.8). This contribute can easily derived by splitting $\{\sigma\}$, $\{\varepsilon\}$ and $[Q]$ in two parts, bending and shear. The founding law (3.8), in the FSDT field, can be re-written as:

$$\begin{Bmatrix} \{N\} \\ \{M\} \\ \{T\} \end{Bmatrix} = \begin{bmatrix} [A] & [B] & [0] \\ [B] & [D] & [0] \\ [0] & [0] & [H] \end{bmatrix} \begin{Bmatrix} \{\varepsilon^0\} \\ \{\kappa\} \\ \{\gamma^0\} \end{Bmatrix} \quad (3.42)$$

where $[H]$ is the shear stiffness matrix.

In the case of uncoupled laminates, the resultants equations can be simplified as follows:

$$\begin{aligned} \{N\} &= [A] \{\varepsilon^0\} \\ \{M\} &= [D] \{\kappa\} \\ \{T\} &= [H] \{\gamma^0\} \end{aligned} \quad (3.43)$$

Stating that:

$$\int_V dV = \int_{-h/2}^{h/2} dz \int_{\Omega} d\Omega = \sum_{k=1}^n \int_{z_k}^{z_{k+1}} dz \int_{\Omega} d\Omega \quad (3.44)$$

and referring to (3.6) and (3.43), the expression (3.41) of the strain energy can be re-written as:

$$\mathcal{E} = \frac{1}{2} \int_{\Omega} \{\varepsilon^0\}^T [A] \{\varepsilon^0\} + \{\kappa\}^T [D] \{\kappa\} + \{\gamma^0\}^T [H] \{\gamma^0\} d\Omega \quad (3.45)$$

Using Voigt's notation introduced in (3.19), the expression (3.45) become:

$$\mathcal{E} = \frac{1}{2} \int_{\Omega} \left\{ \sum_{i=x,y,s} \sum_{j=x,y,s} A_{ij} \varepsilon_i \varepsilon_j + \sum_{i=x,y,s} \sum_{j=x,y,s} M_{ij} \kappa_i \kappa_j + \sum_{i=q,r} \sum_{j=q,r} H_{ij} \gamma_i \gamma_j \right\} d\Omega \quad (3.46)$$

3.5.0.1 Strain Energy Derivatives

The calculation of the energy derivatives is accomplished in the same way by using the derivatives of the homogenised matrices (3.22) whose extended form is given in (B.2).

In FEM field, the governing global equation is written as

$$[K] \{U\} = \{F\} \quad (3.47)$$

where $[K]$ is the global stiffness matrix, given by the assembling of the elements stiffness matrices, $\{F\}$ is the global force vector, given by the assembling of the elements forces and $\{U\}$ is the global displacement vector. Referring to a general derivation parameter v , the derivative of (3.47) with respect to v is:

$$\begin{aligned}\frac{\partial}{\partial v} ([K] \{U\}) &= \frac{\partial}{\partial v} (\{F\}) \\ \frac{\partial}{\partial v} [K] \{U\} + [K] \frac{\partial}{\partial v} \{U\} &= \{0\}\end{aligned}\tag{3.48}$$

The expression of strain energy in the FEM field is

$$\mathcal{E} = \{U\}^T \{F\}\tag{3.49}$$

whose derivative with respect to the general derivation parameter v is

$$\frac{\partial \mathcal{E}}{\partial v} = \frac{\partial}{\partial v} (\{U\}^T \{F\}) = \frac{\partial}{\partial v} (\{U\}^T [K] \{U\}) = \frac{\partial}{\partial v} \{U\}^T [K] \{U\} + \{U\}^T \frac{\partial}{\partial v} ([K] \{U\})\tag{3.50}$$

According to (3.48), the derivative

$$\frac{\partial}{\partial v} (\{F\}) = \{0\}\tag{3.51}$$

allows to write the simplified expression of (3.50)

$$\frac{\partial \mathcal{E}}{\partial v} = \frac{\partial}{\partial v} \{U\}^T [K] \{U\}\tag{3.52}$$

Defining more clearly

$$\frac{\partial}{\partial v} \{U\}^T = \frac{\partial}{\partial v} (\{F\}^T [K^{-1}]^T) = \frac{\partial}{\partial v} ([K]^T \{U\}^T [K^{-1}]^T) = \{U\}^T \frac{\partial}{\partial v} [K] [K]^{-1}\tag{3.53}$$

since $[K] = [K]^T$.

Replacing (3.53) into (3.52) we obtain

$$\frac{\partial \mathcal{E}}{\partial v} = \{U\}^T \frac{\partial}{\partial v} [K] [K]^{-1} [K] \{U\} = \{U\}^T \frac{\partial}{\partial v} [K] \{U\}\tag{3.54}$$

The derivative strain energy expression obtained is particularly interesting: as we know, $[K]$ is defined as follows

$$[K] = \begin{bmatrix} [A] & [B] & [0] \\ [B] & [D] & [0] \\ [0] & [0] & [H] \end{bmatrix}\tag{3.55}$$

which shows that the derivative of $[K]$ is given by the derivatives of the matrices $[A]$, $[B]$, $[D]$ and $[H]$ or their corresponding homogenised.

Regarding this work, the derivatives are computed with respect to the following polar parameters:

- R_{0K}^{A*} , introduced using the relation $R_{0K}^{A*} = (-1)^K R_0^{A*}$,
- R_1^{A*} ,
- Φ_1^{A*} ,

whose complete expressions are given in B.2.

In conclusion, we can state that the derivative of the energy, with respect to a general parameter, can be evaluated by simply deriving the matrices constituting the global stiffness matrix $[K]$.

Chapter 4

Methodology

Unconstrained and *constrained optimization* strategies are many and various. Everyone of those approaches suit better certain problems than others which is why a good knowledge of the problem analysed is needed in order to choose the best solution method.

4.1 Unconstrained Methods

In the field of unconstrained optimization, the logical steps to chase the solution are the same for all the following presented algorithms while the methodology to compute this very solution changes due to different problem approach.

All algorithms for unconstrained minimization require to supply a starting point, which is usually denoted by x_0 . If there is a good knowledge about the application and the data set, the point x_0 may be choose to be a reasonable estimate of the solution. Otherwise, the algorithm must choose the starting point, either by a systematic approach or in some arbitrary manner.

Beginning at x_0 a sequence of iterates is generated by optimization algorithms in the form of $\{x_k\}_{k=0}^{\infty}$ that terminate when either no more progress can be made or when a solution point has been approximated with sufficient accuracy. In order to decide how to move from one iterate x_k to the next and how to perform the value of the current iterate, the algorithms use information about the function f at x_k , and when is possible, also informations from previous iterates $x_0, x_1, \dots, x_{k-1}$. They use these informations to find a new iterate x_{k+1} which presents a lower function value than x_k .

Two fundamental strategies for moving from the current point x_k to a new iterate x_{k+1} were developed: *Line Search* and *Trust Region*.

In the *Line Search* strategy, the algorithm computes a direction p_k and seeks along this direction from the current iterate x_k for a new iterate which presents a lower function value. The distance to move along p_k can be obtained by solving the following one-dimensional minimization problem to find the step length α :

$$\min_{\alpha > 0} f(x_k + \alpha p_k) \tag{4.1}$$

Solving (4.1) exactly, means to obtain the minimum error from the direction p_k but, in spite of this, such approach may be computationally expensive and usually unnecessary. Instead, the line search algorithm generates a limited number of trial step lengths until it finds one that loosely approximates the minimum of (4.1).

In the *Trust Region* strategy, the informations gathered about f are used to build a model function m_k whose behaviour near the current point x_k is approximately the same to that of the objective function f . Since the model m_k may not be a good approximation of the objective function f when x is far from x_k , we limit the search for a minimizer of m_k to some region around x_k . In other words, we find the candidate step p by approximately solving the following subproblem:

$$\min_p m_k(x_k + p) \quad (4.2)$$

where $x_k + p$ lies inside the trust region. If the calculated solution doesn't compute an adequate decrease in f , we conclude that the trust region is too large-the trust region radius is too big-we decrease it and re-solve (4.2).

In conclusion, the line search and trust-region methods differ in the choice of the decreasing direction and in the computation of the move to the next iterate. *Line Search* begins by fixing the direction p_k and then identifying an appropriate step length α_k . In *Trust Region*, the maximum distance is chosen first — the trust-region radius Δ_k — and then the algorithm searches a direction and a step that will allow to obtain the best improvement possible subject to this distance constraint. If the step is proved to be unsatisfactory, the distance measure is reduced Δ_k and the computation performed again.

All the following algorithm can be found in (A).

4.1.1 Line Search Approach

As previously said, each iteration of a line search method computes a search direction p_k and then calculate the advancement along that direction. The iteration is given by the following equation

$$x_{k+1} = x_k + \alpha_k p_k \quad (4.3)$$

where α_k is a positive scalar called the *step length*. The good and successful behaviour of a line search method depends on effective choices of both the direction p_k and the step length α_k . The majority of line search algorithms demand p_k to be a descent direction, that satisfies the property $p^T \nabla f_k < 0$ which assures that the function f can decrease along this direction. Moreover, the search direction is often computes as follow

$$p_k = B_k^{-1} \nabla f_k \quad (4.4)$$

where B_k is a non-singular and symmetric matrix, that changes according to the used algorithm. In the steepest descent method, B_k is the identity matrix I , while in Newton's method, B_k is the exact Hessian $\nabla^2 f_k$. In quasi-Newton methods, B_k is an approximation to the Hessian, updated at every iteration by the evaluation of a low-rank formula. When

p_k is defined by (4.4) and B_k is positive definite, we have

$$p_k^T \nabla f_k = -\nabla f_k^T B_k^{-1} \nabla f_k < 0$$

and consequently p_k is a descent direction.

4.1.1.1 Step Length

The computation of the step length α_k is challenging because on one hand we would like to choose α_k in order to have a significant reduction of f , but on the other hand we don't want to increase the computational cost. The ideal choice would be the global minimizer of the univariate function $\phi(\cdot)$ defined by:

$$\phi(\alpha) = f(x_k + \alpha p_k) \quad \alpha > 0$$

but normally this procedure is too expensive. Typical line search algorithms test a variable number of candidate values for α stopping only when one of these values satisfies certain conditions. The line search is done in two stages: first of all, a bracketing step finds an interval containing desirable step lengths, and then a bisection or interpolation step computes the best step length possible within the interval.

Backtracking Method In unconstrained minimization, a backtracking line search is a line search method to determine the maximum advancement along a given search direction. It starts with a not-too-large estimate of the step size and then decrease iteratively the step size until a reduction of the objective function, that presents similar value to the one anticipated, is observed, based on the local gradient of the objective function.

Wolfe conditions The most conventional inexact line search condition stipulates that α_k should allow a sufficient decrease in the objective function f , according to the inequality:

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T p_k \quad (4.5)$$

for some $c_1 \in (0,1)$. The previous inequality, called the *Armijo condition* or *sufficient decrease condition*, states that the reduction of the objective function f depends to both the step length α_k and the directional derivative $\nabla f_k^T p_k$. In spite of all, equation (4.5) is not enough to guarantee that the algorithm makes reasonable progress because it is satisfied for all sufficiently small values of α . To avoid steps too short a second inequality, called the *curvature condition*, is introduced, which demands α_k to meet

$$\nabla f(x_k + \alpha p_k)^T p_k \geq c_2 \nabla f_k^T p_k \quad (4.6)$$

for some constant $c_2 \in (c_1,1)$. In (4.6) the left-handside is the derivative $\phi'(\alpha_k)$, which means that this second condition ensures that the slope of ϕ at α is greater than c_2 times the initial one $\phi'(0)$. The *sufficient decrease* and *curvature* conditions are known collectively as the *Wolfe conditions*.

Golden Section Method The golden section is a first-order iterative optimization and derivative-free method for finding the minimum of a function. The basic idea for minimizing a unimodal function over $[a, b]$ is iteratively reducing the interval of uncertainty by comparing the function values of the observations. When the length of the interval of uncertainty is reduced to some desired degree, the points on the interval can be regarded as approximations of the minimizer. Such a class of methods only needs to evaluate the function and has wide applications, especially it is suitable to nonsmooth problems and those problems with complicated derivative expressions. The technique derives its name from the fact that the algorithm maintains the function values for triples of points whose distances from a *golden ratio*.

Interpolation Method The interpolation method represents a different approach of line search. This method approximates $\phi(\alpha) = f(x_k + \alpha p_k)$ by fitting a quadratic or cubic polynomial in α to known data and choosing a new value of α which minimizes the polynomial. Then the bracketing interval is reduced by comparing the new value of α and the known points. Usually if the function has good analytical properties, the interpolation method is superior to the golden section method described in the last subsection.

4.1.1.2 Line Search Methods

Gradient Descent Gradient descent, known also as *steepest descent*, is a first-order iterative optimization algorithm. The basis for the method is the simple observation that a continuous function should decrease, at least initially, if one takes a step along the direction of the function negative gradient (or of the approximate gradient) at the current point. The steepest descent direction $-\nabla f_k$ is the most obvious choice for search direction for a line search method because, among all the directions we could move from x_k , it is the one along which f decreases most rapidly.

Newton's Method Newton's Method is a first and second-order iterative optimization algorithm in which, referring to equation (4.4), the direction is computed as follow

$$p_k = -\nabla^2 f_k^{-1} \nabla f_k \quad (4.7)$$

The main issue of Newton's method is that the Hessian matrix $\nabla^2 f_k$ may not be definite positive which could make p_k not to be a descent direction. Since the exact Newton iteration is not guaranteed to produce descent directions when the current iterate is not close to a solution, in order to obtain a globally convergent iteration, the Hessian matrix is modified and approximated to make it positive definite and consequently obtain a consistent descent direction.

Quasi-Newton Methods Quasi-Newton methods are first-order iterative optimization algorithm, which requires only the gradient, like steepest descent method, used as an alternative to Newton's method when the Hessian is unavailable or too expensive to compute

at every iteration. The direction in Quasi-Newton Methods is defined as follow

$$p_k = -B_k^{-1} \nabla f_k \quad (4.8)$$

where the matrix B_k is updated at every iteration by an updating formula. The most-popular Hessian updating formula are the BFGS and the DFP which are described below. The *BFGS method*, named after Broyden, Fletcher, Goldfarb and Shanno, use function values and gradients to build up the Hessian approximation in which the initial approximation of the Hessian is the identity matrix. The new approximated value of the Hessian is given by

$$(BFGS) \quad H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T \quad (4.9)$$

where

$$\rho_k = \frac{1}{y_k^T s_k}, \quad s_k = x_{k+1} - x_k, \quad y_k = \nabla f_{k+1} - \nabla f_k \quad (4.10)$$

The *DFP method*, named after Davidon, Fletcher and Powell, is similar to the BFGS in which at each step the inverse Hessian is updated by the sum of two symmetric rank one matrices.

$$(DFP) \quad H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k} \quad (4.11)$$

In order to strengthen both the algorithms and avoid H_k to become singular, the following *curvature condition* is required

$$s_k^T y_k \geq 0 \quad (4.12)$$

4.1.2 Trust Region Approach

As previously said the Trust Region method, known also as *restricted step method*, define a region around the current iterate and then choose the step to be the approximate minimizer of the model in this region; if the step is not satisfactory the method compute a new radius for the region and find a new value of the objective function. To obtain each step, we seek a solution of the subproblem

$$\min_{p \in \mathbb{R}^n} m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p \quad \|p\| \leq \Delta_k \quad (4.13)$$

where m_k is the *model function*. The first step to compute in a trust-region method is to obtain the trust-region radius Δ_k at each iteration; the choice is based on the agreement between the model and the objective function at the last iteration. Given a step p_k we define

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)} \quad (4.14)$$

where the numerator represents the *actual reduction* and the denominator the *predicted reduction*

4.1.2.1 Subproblem Solution Methods

Eq. (4.13) represents the so-called trust region subproblem that need to be solved in order to compute the step p_k . There are various methodologies implemented to solve (4.13) as the ones presented below. Refer to Appendix A for the following algorithms.

Exact Method The exact method is the simplest approach to solve the sub-problem (4.13) which use the Cholesky factorization $B + \lambda^{(l)}I = R^T R$ to compute an appropriate p_k that presents an high computational cost that can be stopped after two or three iterations.

Cauchy Point Even if we seek for a optimal solution of the subproblem (4.13), it's enough, in order to achieve a global convergence, compute an approximate value of p_k that lies in the trust region and returns a *sufficient reduction* of the model function. This sufficient reduction can be quantified by the Cauchy Point which calculation is not expensive and provide crucial information in deciding the acceptability of an approximate solution. A trust region method will achieve global convergence if the step p_k allows a reduction of the function model hat is at least some fixed positive multiple of the decrease attained by the Cauchy step.

Dogleg and Double Dogleg Methods Since the Cauchy point method only implements a gradient descent method-which performs poorly in some cases- with a particular choice of step length we can improve the search of the approximate solution of (4.13). The *dogleg method* represents the first improved approach based on including the curvature information from B_k and substituting the curvilinear path of p with a two-segment approximation.

The *double dogleg method* combines the ideas of quasi-Newton and trust region methods and computes in each iteration the step p_k as the linear combination of the steepest descent direction and a quasi-Newton search direction; this method works well for medium to moderately large optimization problems where the objective function and the gradient are much faster to compute than the Hessian or one of its approximations.

4.2 Constrained Methods

In this section, some constrained optimization algorithms are presented. We don't presume to be exhaustive because this research field is worldwide very prolific of new ideas and applications but, nevertheless, the basic ideas constituting the background for the most part of the known constrained optimization algorithms are illustrated. In the field of constrained optimization it is possible to identify three different kind of problems: linear, quadratic and sequential quadratic. *Linear programs* have a linear objective function and linear constraints, which may include both equalities and inequalities; *quadratic programs* have a quadratic function of several variables subject to linear constraints while *sequential quadratic programs* have both quadratic function and constraints. First of all we can identify two different approaches in constrained optimization problems solving: the first one,

always referring to (2.1), is to transform the constrained problem into an unconstrained one, and then use an unconstrained algorithm, such as Newton or Quasi-Newton, to solve it, while the second one consists in solving successive constrained quadratic subproblems.

4.2.1 Linear Programming

In linear programming there are two main methods used to solve the constrained problem, by approximating an unconstrained equivalent one: the *Penalty method* and the *Lagrangian method*.

Penalty Method

The penalty method replaces a constrained optimization problem by a series of unconstrained problems whose solutions ideally converge to the solution of the original constrained problem. The Penalty method or *Quadratic Penalty* method, defines individual terms for the constraints that are multiplied by a positive coefficient; by making this coefficient larger, we penalize constraint violations, thereby forcing the minimizer of the penalty function closer to the feasible region for the constrained problem. Referring to (2.1) the corresponding penalty function can be written as follows:

$$Q(x, \varepsilon) = f(x) + \frac{\varepsilon}{2} \sum_{i \in \mathcal{E}} c_i^2(x) + \frac{\varepsilon}{2} \sum_{j \in \mathcal{I}} (\max\{-c_j(x), 0\})^2 \quad (4.15)$$

where $\varepsilon > 0$ is the *penalty parameter*. In order to penalize the constraint violations with increasing severity, we drive $\varepsilon \rightarrow \infty$.

Lagrangian Method

In general, the Lagrangian is the sum of the original objective function and a term that involves the functional constraint and the so called *Lagrangian Multiplier* λ . Referring again to (2.1), the Lagrangian function associated is:

$$L_a(x, \lambda, \mu) = f(x) - \sum_{i \in \mathcal{E}} \lambda_i c_i(x) - \sum_{j \in \mathcal{I}} \mu_j c_j(x) \quad (4.16)$$

where $\boldsymbol{\lambda} = (\lambda_i)$ and $\boldsymbol{\mu} = (\mu_i)$ are the *Lagrangian multipliers* associated with equality and inequality constraints. In order to define x^* a minimizer for (2.1) problem the following conditions must be satisfied:

$$\begin{aligned} \nabla_x L(x^*, \lambda^*, \mu^*) &= \nabla f(x^*) - \sum_{i \in \mathcal{E}} \lambda_i \nabla c_i(x) - \sum_{j \in \mathcal{I}} \mu_j \nabla c_j(x) = 0, \\ c_i(x^*) &= 0 \quad \forall i \in \mathcal{E}, \\ c_j(x^*) &= 0 \quad \forall j \in \mathcal{I}, \\ \mu^* &\geq 0 \quad \forall j \in \mathcal{I}, \\ \mu^* c_j(x^*) &= 0 \quad \forall j \in \mathcal{I} \end{aligned} \quad (4.17)$$

The previous conditions are called the *Karush-Kuhn-Tucker conditions* and a point x^* is called a *KKT point* for L if there exist λ^* and μ^* such that (x^*, λ^*, μ^*) satisfy (4.17).

4.2.2 Quadratic Programming

Speaking of quadratic programming, the approach is the same as the linear programming but where, some modification are introduced. While the Quadratic Penalty method presents the same expression, certain changes are introduced in the Lagrangian method, which is now called the *Augmented Lagrangian* method.

Augmented Lagrangian Method

The Augmented Lagrangian is related to the Quadratic Penalty previously described but, unlike the penalty approximation, this approach preserves smoothness and reduces the ill conditioning. Keeping referring to (2.1), and taking into account only the equality constraints, the augmented Lagrangian function is

$$L_a(x, \lambda, \varepsilon) = f(x) - \sum_{i \in \mathcal{E}} \lambda_i c_i(x) + \frac{\varepsilon}{2} \sum_{i \in \mathcal{E}} c_i^2(x) \quad (4.18)$$

where we see that the augmented Lagrangian differs from the (standard) Lagrangian (4.16) by the presence of the squared terms, while it differs from the quadratic penalty function (4.15) in the presence of the summation term involving λ . In this sense, it is a combination of the Lagrangian function and the Quadratic Penalty function. In presence of both equality and inequality constraints, the Augmented Lagrangian function is defined as follows

$$L_a(x, \lambda, \mu, \varepsilon) = f(x) + \sum_{i \in \mathcal{E}} \lambda_i c_i(x) + \frac{\varepsilon}{2} \sum_{i \in \mathcal{E}} c_i^2(x) + \sum_{j \in \mathcal{I}} \mu^T \max \left\{ c_j(x), -\frac{\varepsilon}{2} \mu \right\} + \frac{\varepsilon}{2} \sum_{j \in \mathcal{I}} \max \left\{ c_j(x), -\frac{\varepsilon}{2} \mu \right\}^2 \quad (4.19)$$

4.2.3 Sequential Quadratic Programming

For reason of clarity, before introducing the SQP method, a brief introduction of the term *active set* is provided to the reader. Accordingly to [17] and (2.1), the active set $A(x)$ at any feasible x consists of the equality constraint indices from $\mathcal{E}$ together with the indices of the inequality constraints i for which $c_i(x) = 0$; that is,

$$A(x) = \mathcal{E} \cup \{i \in \mathcal{I} \mid c_i(x) = 0\}$$

At a feasible point x , the inequality constraint $i \in \mathcal{I}$ is said to be *active* if $c_i(x) = 0$ and *inactive* if the strict inequality $c_i(x) > 0$ is satisfied. The active set at x consists of those constraints $c_i(x)$ that are active at the current point.

A general active set method based starts by making a guess of the optimal active set A , that is, the set of constraints that are satisfied as equalities at a solution. We call our guess the *working set* and denote it by W . We then solve a problem in which the constraints in the working set are imposed as equalities and the constraints not in W are ignored. We then check to see if there is a choice of Lagrange multipliers such that the solution x obtained for this W satisfies the KKT (4.17). If so, we accept x as a local solution of

(2.1). Otherwise, we make a different choice of W and repeat the process. This approach is based on the observation that, in general, it is much simpler to solve equality-constrained problems than to solve nonlinear programs.

The Sequential Quadratic Programming (SQP) methods are mostly active set based and can be considered as an extension of Newton's method to the constrained field. The basic idea is to move away from the current point by minimizing a quadratic model of the problem. Unlike the Penalty and Lagrangian methods previously described, which are mostly effective when the constraints are linear, the SQP methods are succesful when significant nonlinearities are present in the constraints; they can be used both in line search and trust-region frameworks and are appropriate for small or large problems.

Using $A(x)$ to denote the Jacobian matrix of the constraints, we can write the KKT conditions of the constrained problems as a system of $n+m$ equations in the $n+m$ unknowns x and λ :

$$F(x, \lambda) = \begin{bmatrix} \nabla f(x) - A(x)^T \lambda \\ c(x) \end{bmatrix} \quad (4.20)$$

whose Jacobian is

$$F'(x, \lambda) = \begin{bmatrix} \nabla_{xx}^2 L(x, \lambda) & -A(x)^T \\ A(x) & c(x) \end{bmatrix} \quad (4.21)$$

The Newton step from iterate (x_k, λ_k) is given by

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix} \quad (4.22)$$

where p_k and p_λ solve the KKT system

$$\begin{bmatrix} \nabla_{xx}^2 L(x, \lambda) & -A(x)^T \\ A(x) & 0 \end{bmatrix} \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix} = \begin{bmatrix} \nabla f(x) - A_k^T \lambda_k \\ -c_k \end{bmatrix} \quad (4.23)$$

4.3 B-spline Curves and Surfaces

In this section, the fundamentals of the B-Spline Curves and Surfaces theory are briefly recalled, since the VAT composite plate case study is performed using this particular mathematical framework. A B-spline curve is a generalization of the Bézier curve

$$C(u) = \sum_{i=0}^n B_{i,n}(u) P_i \quad 0 \leq u \leq 1 \quad (4.24)$$

where the basis functions B_i are the classical n th-degree Bernstein polynomials. The geometric coefficients of this polynomial are the so called *control points* $\{P_i\}$, while the places where the polynomials meet are known as *knots*. Given $n+1$ control points $P_0, P_1, \dots, P_n$ the B-spline curve of degree p defined by these control points and knot vector U is

$$C(u) = \sum_{i=0}^n N_{i,p}(u) P_i \quad 0 \leq u \leq 1 \quad (4.25)$$

where $N_{i,p}(u)$ are standard blending functions of degree p . According to the notation of [15], a B-spline surface is defined as

$$S(u, v) = \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} N_{i,p}(u) N_{j,q}(v) P_{i,j} \quad (4.26)$$

where u and v are scalar dimensionless parameters both defined in the interval $[0,1]$, p and q are the B-spline degrees along u -direction and v -direction, respectively, and $\mathbf{P}_{i,j} = \{x_{i,j}, y_{i,j}, z_{i,j}\}$ are the Cartesian coordinates of the control points in the form of Greville abscissae [14]. The net of $(n_u + 1) \times (n_v + 1)$ control points constitutes the so-called *control net*.

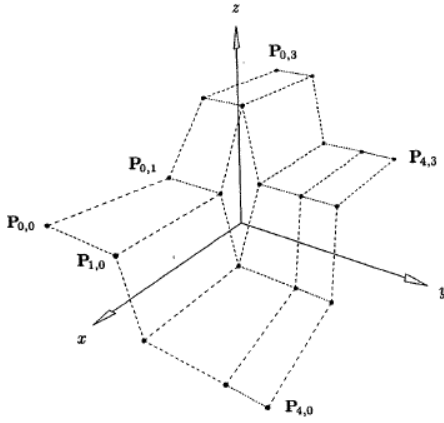


Figure 4.1: B-Spline Surface Control Net

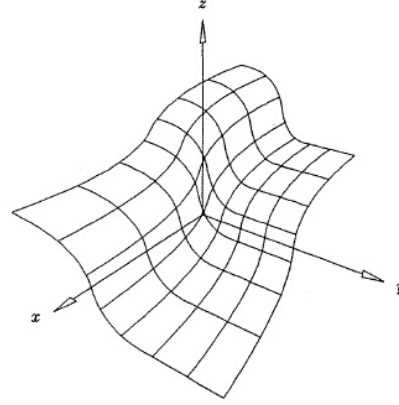


Figure 4.2: B-Spline Surface

The blending functions are defined recursively by means of the Bernstein polynomials as follows:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } U_i \leq u \leq U_{i+1}, \\ 0 & \text{otherwise} \end{cases}, \quad (4.27)$$

$$N_{i,p}(u) = \frac{u - U_i}{U_{i+p} - U_i} N_{i,p-1}(u) + \frac{U_{i+p+1} - u}{U_{i+p+1} - U_{i+1}} N_{i+1,p-1}(u)$$

where U_i is the i -th component of the non-periodic non-uniform *knot vector*:

$$\mathbf{U} = \left\{ \underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{1, \dots, 1}_{p+1} \right\} \quad (4.28)$$

The size of the knot vector is $m_u + 1$ where

$$m_u = n_u + p + 1$$

Analogously, the $N_{j,q}(v)$ are defined on the knot vector $\mathbf{V}$, whose size is m_v :

$$\mathbf{V} = \left\{ \underbrace{0, \dots, 0}_{q+1}, v_{q+1}, \dots, v_{s-q-1}, \underbrace{1, \dots, 1}_{q+1} \right\} \quad (4.29)$$

$$m_v = n_v + q + 1$$

The knot vectors $\mathbf{U}$ and $\mathbf{V}$ are two non-decreasing sequences of real numbers that can be interpreted as two discrete collections of values of the dimensionless parameters u and v . As the control points, also the knot vectors components form a net. One basic property of the blending functions is the *local support property*: $N_{i,p}(u) = 0$ if u is outside the interval $[U_i, U_{i+p+1})$; this property is likewise valid for $N_{j,q}(v) = 0$ on $[V_j, V_{j+q+1})$. For a deeper insight in the B-Spline Curves and Surfaces theory, the reader is addressed to [15].

4.4 Matlab `fmincon`

According to MathWorks Support definition, `fmincon` is a non-linear programming solver to find minimum of constrained nonlinear multivariable function. The input required by this function are:

- x_0 is the initial point, specified as a real vector or real array;
- f is the function to minimize, specified as a function handle or function name;
- c_i linear and non-linear constraints specified as
 - *matrices* if linear
 - *function handle* or *function name* if non-linear

The algorithms implemented are:

- Interior Point;
- Trust Region Reflective;
- SQP;
- Active Set;

Every FMINCON algorithm allow the user to provide the gradient for the objective function and for the non-linear constraints; therefore, Trust Region and Interior Point Method allow to provide the Hessian matrix of the objective function too. Regarding this works, the algorithm used is the Active Set, by providing in addition the constraints and function gradients.

4.5 Genetic Algorithm BIANCA

Genetic Algorithms represent a special class of evolution-based systems, often referred as Evolution Programs [7]. A genetic algorithm (GA) is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. The algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution. The architecture of a standard GA is give in Figure 4.3.

The standard GA is composed by the union of 3 fundamental operators:

- the *selection* operator,
- the *crossover* operator,
- the *mutation* operator.

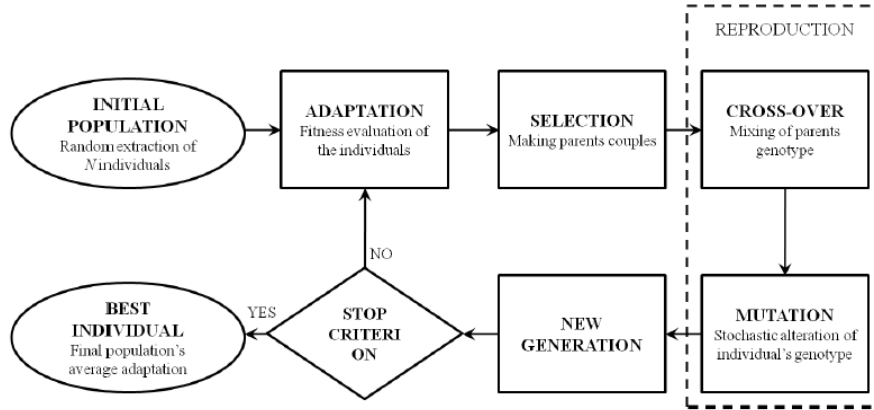


Figure 4.3: The architecture of the standard GA

The selection operator is an operator that acts in the following way: considering a N_{ind} individuals population and using the value of the *fitness function* of each individual, the selector operator selects, with a higher probability, the individuals having a high value of the fitness function. The fitness function gives a numerical value at each individual-point of the design space, and consequently the most adapted individuals (i.e. points which are candidates to be potential optimal solutions) will be the points having higher values of the fitness function. After assigning a fitness value to each individual of the population, the selection operator determines which individuals will take part into the real reproduction process, which will leads to the creation of the new generation of individuals.

The crossover operator is the operator that concretely achieves the creation of new individuals with a process articulated in two steps:

- the N_{ind} individuals are coupled forming the couples of parents,
- every single gene of each chromosome of the individual's genotype is randomly cut, in one or more locations (the same positions for each homologous gene of the couple): at this point two new individuals are created by mixing and crossing the information.

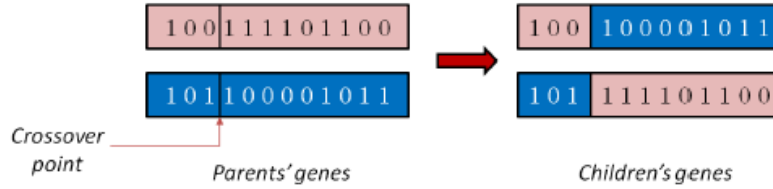


Figure 4.4: Effect of the crossover operator on two homologous genes of the parents' couple.

The mutation operator acts in a random way on the structure of the individuals at the level of the genes of the new individuals generated after the crossover phase. The mutation operator works on the single bit of the chain, by switching it from 0 to 1 or vice-versa. The main aim of the mutation process consists in increasing biodiversity among the individuals composing the population. Introducing and increasing biodiversity, through the mutation mechanism, is a crucial point in the GA search process: in fact, through the biodiversity it is possible to avoid a premature convergence of the algorithm towards local minima and/or pseudo-optimal solutions.

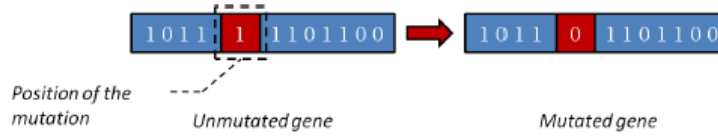


Figure 4.5: Effect of the mutation operator on the bits of the single gene.

For a deeper insight in the Genetic Algorithm BIANCA theory, the reader is addressed to [7].

4.6 ANSYS Mechanical APDL

The numerical analysis of both study cases of this thesis is carried out in ANSYS Mechanical, where several codes have been programmed in the ANSYS Parametric Design Language (APDL). The general procedure for both the problems is divided in two main stages, an initial and a final one. In the initial stage the geometry and the boundary condition of the plate are defined, as well as the values of the constant polar parameters

while, the polar parameters which represent the optimization variables, firstly are given as dimensionless values by a random selection, subjected to a volume constraint, performed by MATLAB function `rand` and after as the output of the genetic algorithm BIANCA. The physical loads and constraints of the plate are defined in this stage as well. In the final stage the evaluation of the energy and its derivatives with respect to the polar parameters R_0 , R_1^{A*} and Φ_{1A*} is performed by using several macro implemented by the author.

4.6.1 Pre Integrated Matrices

Composite materials usually present different values of thickness and orientation for each layer. In the environment of ANSYS Mechanical APDL, the definition of the laminate properties can be accomplished in two different ways:

- by providing directly the material properties as Young Modulus, Poisson Ratio, Shear Modulus, etc. ;
- by providing directly the structural matrices describing the laminate.

Within the framework of this work, the polar parameters are used in order to describe the laminate; this leads, in order to perform the analysis with ANSYS Mechanical APDL, to the use of Pre Integrated Matrices built as functions of the polar parameters as described in section (3.4.1). In particular, as regards this works, the homogenised matrices given in (B.1) are used because, as previously said, the best practice, in order to analyse the elastic response of the multilayer plate, is the introduction of the homogenised structural matrices dependent on the polar parameters.

Chapter 5

Study cases

The purpose of this work is to minimize the strain energy of an elementary multilayered composite plate. In order to achieve this goal, two different study cases are taken into account: the first study case focuses on the Unidirectional Fiber composites field while the second study case takes into account the study of VAT composites one.

Unidirectional fiber distribution means that constant values of the polar parameters R_{0K}^{A*} , R_1^{A*} and Φ_1^{A*} are taken into account and kept fixed throughout the whole plate, while a VAT fiber distribution presents several values of the polar parameters, defined as unknowns in certain points of the plate.

Regarding the analysis, the two studies were firstly performed by using ANSYS Mechanical APDL and MATLAB function `fmincon`, then, for reasons subsequently explained, with the genetic alghoritm BIANCA, before repeating the first procedure.

5.1 Description of the problem

Concerning the study cases here presented, the optimization strategy is applied to a unidirectional laminated plate as well to a VAT laminate plate. Both plates have a fixed total thickness and are made of a carbon-epoxy pre-preg strips whose properties, in terms of technical constants and polar parameters as well, are listed in Table 5.1. Regarding the values of R_{0K}^{A*} , R_1^{A*} and Φ_1^{A*} , the lower and upper bounds are given in the respectively study case sections along the geometrical characteristics.

Technical constants		Polar parameter of $[Q]$		Polar parameter of $[\hat{Q}]$	
E_1 [MPa]	161000.0	T_0 [MPa]	23793.3868	T [MPa]	5095.4545
E_2 [MPa]	9000.0	T_1 [MPa]	21917.8249	R [MPa]	1004.5454
G_{12} [MPa]	6100.0	R_0 [MPa]	17693.3868	Φ [deg]	90.0
ν_{12}	0.26	R_1 [MPa]	19072.0711		
ν_{23}	0.10	Φ_0 [deg]	0.0		
		Φ_1 [deg]	0.0		

Table 5.1: Material properties of the carbon-epoxy pre-preg strip taken from [10].

About the mechanical behaviour, the following hypotesis [10] have been made:

- the geometry and the BCs of the structure are know and fixed;
- the plates are made of identical plies;
- the material presents a linear elastic behaviour;
- the properties are uniformly distributed throughout the unidirectional fiber plate while apply locally in each point of the structure in the VAT plate;
- the elastic responses of the plates are described in the theoretical framework of the FSDT and the stiffness matrices ar expressed in terms of the laminate polar parameter, as showed in 3.4.1.

5.2 Matemathical Statement of the Problem

The problem focuses on the definition of the optimal distribution of the laminate polar parameters in order to achieve the minimization of the strain energy. In this background, the solution of the structural optimisation problem is searched for an orthotropic and quasi-homogeneous plate subject to given BCs.

Therefore the optimisation problem can be formulated as follows:

$$\begin{aligned} & \min_x \quad \mathcal{E}(\mathbf{x}) \\ & \text{subject to} \\ & g(\mathbf{x}) \leq 0 \end{aligned} \tag{5.1}$$

where $\mathcal{E}(\mathbf{x})$ is the strain energy.

Regarding this work, a volume constraint is taken into account, which means that, in addition of the formulation of the optimisation problem , the geometric and feasibility constraints on the polar parameters (which arise from the combination of the layers orientations and positions within the stack) must also be considered. This constraint ensure that the optimum values of the polar parameters resulting from the first step correspond to a feasible laminate that will be designed during the second step of the optimisation strategy. Since the laminate is quasi-homogeneous, such constraints can be written only for matrix $[A^*]$ as follows:

$$\left\{ \begin{array}{l} -R_0 \leq R_{0K}^{A^*} \leq R_0, \\ 0 \leq R_1^{A^*} \leq R_1, \\ 2 \left(\frac{R_1^{A^*}}{R_1} \right)^2 - 1 - \left(\frac{R_{0K}^{A^*}}{R_0} \right) \leq 0 \end{array} \right. \tag{5.2}$$

First and second constraints of 5.2 can be taken into account as admissible intervals for the relevant optimisation variables, i.e. on $R_{0K}^{A^*}$ and $R_1^{A^*}$. Hence, the resulting feasibility constraint on the laminate polar parameters is:

$$g(\mathbf{x}) = 2 \left(\frac{R_1^{A^*}}{R_1} \right)^2 - 1 - \left(\frac{R_{0K}^{A^*}}{R_0} \right) \leq 0. \tag{5.3}$$

5.3 Preliminary Analysis

As previously said, in the first analysis, the minimization of the strain energy is performed by interfacing ANSYS Mechanical APDL and MATLAB function `fmincon`. The analysis requires a main MATLAB code and an ANSYS programming file, as well as several sub-routines. The main MATLAB code carries out the random generation of the initial point and the calling of `fmincon` and it's supported by the secondaries scripts of the objective function and constraint function respectively. The main ANSYS program file uses four different macros to evaluate the strain energy and its derivatives; those macros calculate respectively the general value of the strain energy, and the three derivatives of the homogenized matrices $[A^*]$, $[D^*]$ and $[H^*]$ with respect to the polar parameter $R_{0K}^{A^*}$, $R_1^{A^*}$ and $\Phi_1^{A^*}$, following used to calculate the energy derivatives with respect to the same polar parameter, according to the explanation in chapter 3.5. The MATLAB code receives the objective function value in the initial point, calculated by the `rand` function, and the gradients' values of the energy derivatives with respect to the polar parameter $R_{0K}^{A^*}$, $R_1^{A^*}$ and $\Phi_1^{A^*}$ from the ANSYS program file; after it accomplishes a minimization with `fmincon` reducing the strain energy value at each iteration. This first analysis brought out that the problems taken into account aren't strongly convex problems. This means that, as shown in the results, the solver stopped to minimize the function both after finding a local minimum or after reaching the maximum number of iterations and the minimum strain energy value returned by `fmincon` it's different for different starting points.

5.4 Unidirectional Fiber Composite Plate

The first study case implemented presents two analysis of a quasi-homogenised unidirectional fiber composite fixed-ended plate subjected to a volume constraint: in the first analysis the plate is subjected to a tensile stress of 100N while in the second analysis the plate is subjected to a bending stress of 50N. The FE model is built within the ANSYS Mechanical environment and is made of SHELL181 elements based on the Reissner-Mindlin kinematic model, having 4 nodes and six Degrees Of Freedom (DOFs) per node. The geometry data of the plate and the BCs are given in Table 5.2 and Table 5.3 respectively.

Length [mm]	Width [mm]	Thickness [mm]
360	140	3

Table 5.2: Unidirectional Fiber Composite Plate Dimensions

Sides	BCs
AD	$U_x, U_y, U_z = 0$ $\varphi_x, \varphi_y, \varphi_z = 0$
AB,BC,CD	free

Table 5.3: BCs of the Unidirectional Fiber Composite Plate

The lower and upper bounds of the mechanical design variables R_{0K}^{A*} , R_1^{A*} and Φ_1^{A*} [10], in the case of the unidirectional fiber composite plate are listed in Table 5.4.

Design Variable	Type	Lower bound	Upper bound
R_{0K}^{A*}	Continuous	-17693.3868	17693.3868
R_1^{A*}	Continuous	0.0	19072.0711
Φ_1^{A*}	Continuous	-90.0	90.0

Table 5.4: Design space of the Unidirectional Fiber Composite Plate problem

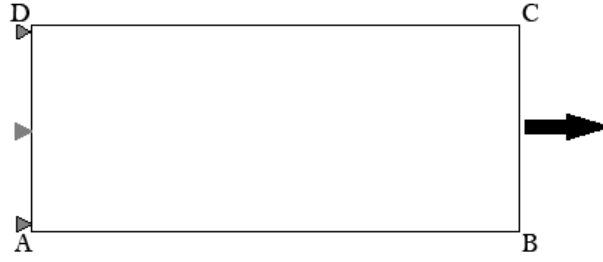


Figure 5.1: Unidirectional fiber composite plate BCs

5.4.1 Simple Plate subjected to tensile stress

The model of the problem is shown in Figure 5.2.

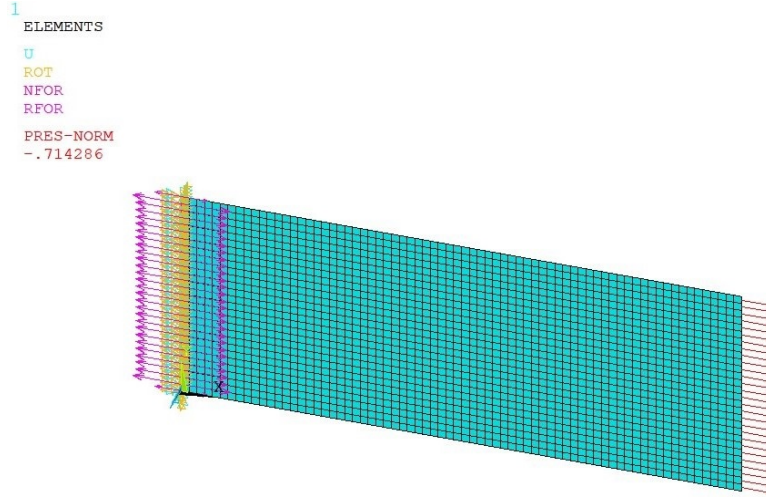


Figure 5.2: Plate subjected to tensile stress model

As previously said, since the problems is not strongly convex, which means that multiple local minimum are present, the strain energy minimum in the function minimizer differs, depending on the initial point, as shown in Table 5.5.

Case	Iteration	Func. Eval.	R_{0K}^{A*}	R_1^{A*}	Φ_1^{A*}
I	150	2387	0.826752	0.632359	-0.746026
II	63	983	-0.804919	0.278498	0.093763
III	150	2401	0.915014	0.964888	-0.684774
IV	150	2401	0.941186	0.957167	-0.029249

Table 5.5: MATLAB `fmincon` input and output parameters

Case	R_{0k}	R_1^{A*}	Φ_{1A}^{A*}	S.E.
I	0.836485	0.654227	-0,064563	0.003664
II	-0.660452	0.266998	0.356304	0.005335
III	0.914285	0,964573	-0.681874	0.034824
IV	0.941293	0.957162	-0.028476	0.002846

(a) Adimensional values of local minimum

Case	$R_{0k}[MPa]$	$R_1^{A*}[MPa]$	$\Phi_{1A}^{A*}[^\circ]$	S.E.[$N * mm$]
I	14800.2643	12477.4827	-5.81070	0.036643
II	-11685.643	5092.02814	32.06740	0.053350
III	16176.7987	18396.4127	-61.3687	0.348243
IV	16654.6701	18255.0592	-2.56288	0.028464

(b) Dimensional values of local minimum

Table 5.6: Results for the unidirectional fiber composite plate traction analysis performed with ANSYS APDL and MATLAB `fmincon`

5.4.2 Simple Plate subjected to bending stress

The model of the problem is shown in Figure 5.3.

Case	Iteration	Func. Eval.	R_{0K}^{A*}	R_1^{A*}	Φ_1^{A*}
I	150	2388	0.826752	0.632359	-0.746026
II	150	2388	-0.804919	0.278498	0.0937630
III	150	2401	0.915014	0.964888	-0.684774
IV	150	2401	0.941186	0.957167	-0.029249

Table 5.7: MATLAB `fmincon` input and output parameters

Even in this case, the problem is not strongly convex, so the strain energy minimum in the function minimizer differs depending on the initial point, as shown in Table 5.7.

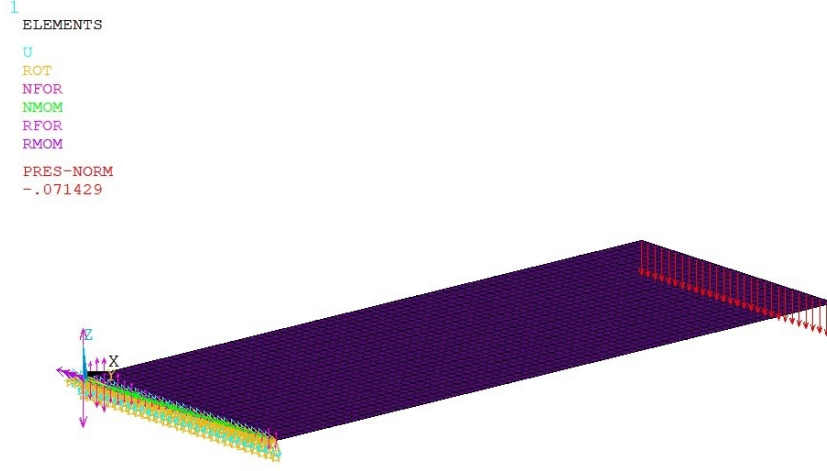


Figure 5.3: Bending plate model

Case	R_{0k}	R_1^{A*}	Φ_{1A}^*	S.E.
I	0.906010	0.816583	0.124673	0.184091
II	-0.902457	0.158075	-0.45288	0.238652
III	0.914255	0.964566	-0.681875	1.524272
IV	0.942872	0.957524	-0.026928	0.130706

(a) Adimensional values of local minimum

Case	$R_{0k}[MPa]$	$R_1^{A*}[MPa]$	$\Phi_{1A}^*[^{\circ}]$	S.E. $[N * mm]$
I	16030.34455	15573.93012	11.22064	552.27
II	-15967.5268	3014.82180	-40.7588	715.956
III	16176.26616	18396.26337	-61.3687	4572.81
IV	16682.60647	18261.97336	-2.42357	392.118

(b) Dimensional values of local minimum

Table 5.8: Results for the unidirectional fiber composite plate traction analysis performed with ANSYS APDL and MATLAB `fmincon`

5.5 Unidirectional Fiber Composite Hole Plate

In this section, the same analysis of 5.4 were performed with the same plate model used hereinafter in Chapter 5.6 in order to have a full comparison, in all fairness, of the results obtained. The initial points are the same as in the Chapter 5.4 to provide the reader an additional comparison with the previously results as well. The geometry data of the plate and the BCs are given in Table 5.9 and 5.10 while the design space of the problem is settled as same as Table 5.4. The two load configuration are the same as Chapter 5.4, as well as the FE model elements. Both the dimensional and adimensional values, which are

used by **fmincon** to perform the analysis, of all the polar parameters, i.e. the optimisation variables, and the strain energy are here provided. The values of the polar parameters are valid only associated to the local strain energy minimum returned by **fmincon**, since, as said, the problem analysed is not strongly convex.

Length [mm]	Width [mm]	Thickness [mm]	Radius[mm]
330	110	3	25

Table 5.9: Unidirectional Fiber Composite Hole Plate Dimensions

Sides	BCs
AD	$U_x, U_y, U_z = 0$ $\varphi_x, \varphi_y, \varphi_z = 0$
AB,BC,CD	free

Table 5.10: BCs of the Unidirectional Fiber Composite Hole Plate

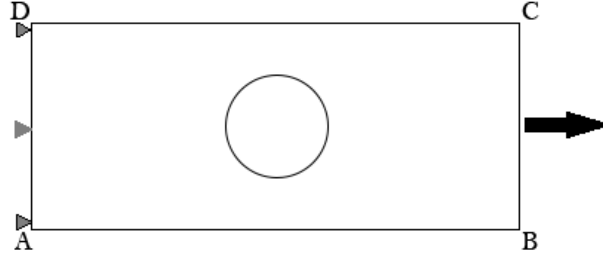


Figure 5.4: Unidirectional Fiber Composite Hole Plate BCs

5.5.1 Hole Plate subjected to tensile stress

The model of the problem is shown in Figure 5.5.

Case	Iteration	Func. Eval.	R_{0K}^{A*}	R_1^{A*}	Φ_1^{A*}
I	150	2387	0.826752	0.632359	-0.746026
II	150	2363	-0.804919	0.278498	0.093763
III	150	2401	0.915014	0.964888	-0.684774
IV	150	2401	0.941186	0.957167	-0.029249

Table 5.11: MATLAB **fmincon** input and output parameters

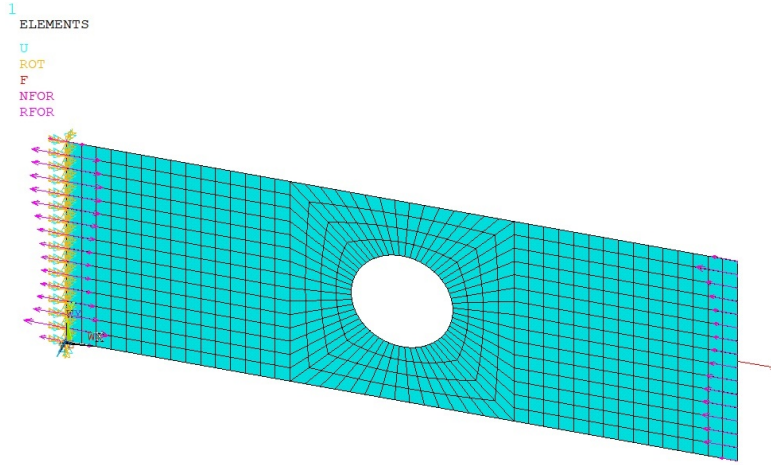


Figure 5.5: Hole plate traction model

Case	R_{0k}	R_1^{A*}	Φ_{1A}^*	S.E.
I	0.836144	0.660571	0.036537	0.005218
II	0.137766	0.268899	0.091363	0.007437
III	0.914264	0.964568	-0.681874	0.047743
IV	0.941638	0.957478	-0.020400	0.004503

(a) Adimensional values of local minimum

Case	$R_{0k}[MPa]$	$R_1^{A*}[MPa]$	$\Phi_{1A}^*[^{\circ}]$	S.E.[$N * mm$]
I	14794.21678	12598.45788	3.28833	0.052177
II	2437.54912	5128.47877	8.22271	0.074380
III	16176.43086	18396.30999	-61.36872	0.477430
IV	16660.77979	18261.09865	-1.83606	0.045038

(b) Dimensional values of local minimum

Table 5.12: Results for the unidirectional fiber composite hole plate traction analysis performed with ANSYS APDL and MATLAB `fmincon`

5.5.2 Hole Plate subjected to bending stress

The model of the problem is shown in Figure 5.6.

Case	Iteration	Func. Eval.	R_{0K}^{A*}	R_1^{A*}	Φ_1^{A*}
I	400	6388	0.826752	0.632359	-0.746026
II	400	6349	-0.804919	0.278498	0.0937630
III	400	6401	0.915014	0.964888	-0.684774
IV	400	6401	0.941186	0.957167	-0.029249

Table 5.13: MATLAB `fmincon` input and output parameters

Case	R_{0k}	R_1^{A*}	Φ_{1A}^*	S.E.
I	0.906010	0.816583	0.124673	0.184091
II	−0.902457	0.158075	−0.45288	0.238652
III	0.914255	0.964566	−0.681875	1.524272
IV	0.942872	0.957524	−0.026928	0.130706

(a) Adimensional values of local minimum

Case	$R_{0k}[MPa]$	$R_1^{A*}[MPa]$	$\Phi_{1A}^*[^{\circ}]$	S.E.[$N * mm$]
I	16030.34455	15573.93012	11.22064	552.27
II	−15967.5268	3014.82180	−40.7588	715.956
III	16176.26616	18396.26337	−61.3687	4572.81
IV	16682.60647	18261.97336	−2.42357	392.118

(b) Dimensional values of local minimum

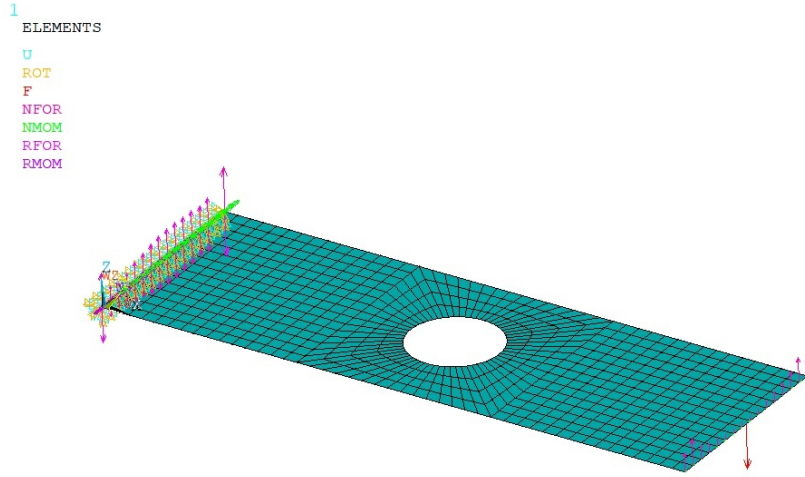
Table 5.14: Results for the unidirectional fiber composite hole plate bending analysis performed with ANSYS APDL and MATLAB `fmincon`

Figure 5.6: Hole plate bending model

For sake of clarity, it should be specified that the assial violet arrows displayed in both Figure 5.5 and Figure 5.6 are due to the use of a *master node* in order to apply the tensile force, and bending as well, insted of applying a line force, as it's done in Chapter 5.4.

This was done because of some errors displayed during some test analysis, likely due to the use of the polar parameters approach to assemble the pre-integrated matrices given by the ANSYS Mechanical model.

It can be noticed that, regarding the analysis performed for the hole plate, the results show that even this problem is not strongly convex, which means again that the strain energy minimum in the function minimizer differs depending on the initial point. In both

configurations the solver stopped because of the maximum number of iteration reached or because a local minimum was found.

5.6 VAT laminate plate

Concerning the VAT analysis, the involvement of BIANCA is due to the fact that the problem of the strain energy minimization is not strongly convex, as highlighted in sections 5.4 and 5.5. The procedure adopted acts according to a two-level optimisation strategy, which is based on two steps: in order to choose a feasible starting point, a first analysis with the Genetic Algorithm BIANCA is performed, then, an optimisation with MATLAB function `fmincon` is carried out with the initial point returned by BIANCA. The initial point returned by BIANCA, consists of a various number of polar parameter adimensional values, respectively, R_{0K}^{A*} , R_1^{A*} and Φ_1^{A*} . In addition to the Genetic Algorithm, the VAT hole plate study case, take into account the B-Spline Surfaces issue too, whose mathematical background is given in section 4.3. A brief introduction to the modus operandi followed by BIANCA to calculate the initial point and to the B-Spline design, is here given below; for a deeper insight in the Genetic Algorithm BIANCA methodology and B-Spline surfaces, the reader is addressed to [16] and [15] respectively.

5.6.1 BIANCA Environment

There are two different kind of inputs for BIANCA. In particular, the main inputs for the code are written in two input files with extension *.gen* and *.opt* respectively; the first input file contains the genetic parameters of the simulation, whilst the second contains the optimisation parameters. The *.gen* file structure is designed as an item of values whose the most relevant are defined below:

- number of population n_{pop} ;
- number of individuals n_{ind} ;
- stop criterion *stop crit* which could be the number of generation, the finding of the individual which satisfy the sill value on the objective function or a combination of the both;
- crossover probability p_{cross} ;
- mutation probability p_{mut} ;
- shift operator probability p_{shift} ;
- I_{time} , represents the number of generation during which the populations are isolated. Every I_{time} generations an exchange of the best feasible individuals among the populations is realised.

Within the framework of this work, the previous parameters are given in Table 5.15

Parameters	Value
n_{pop}	2
n_{ind}	120
$stop\ crit$	FIXED GENERATION
p_{cross}	0.85
p_{mut}	0.008
I_{time}	10

Table 5.15: Input parameters for *.gen* file GA BIANCA

The input file with extension *.opt* contains the optimisation parameters of the simulation. The most relevant structure items of the file are defined below:

- MODEL NAME, name of the file;
- MODEL I, name of the input file;
- MODEL O, name of the output file;
- MAXORMIN, ID for maximisation or minimisation;
- n_{obj} , number of objective function;
- CONSTR, ID for constraints;
- n_{constr} , number of constraints.

At the end of the optimisation process, three output files, with three different extension, are returned, filled with different information. The *.bio* file contains the informations about the best feasible individual for every generations; the *.pop* contains the informations about the whole population every 10 generations while the *.sta* file contains the informations about the statistics on the whole population for each generation. The data necessary for this work have been taken from the *.bio* file and handled as explained in the following sections.

5.6.2 B-spline Domain

In order to introduce the B-spline framework into the study case take into account the following procedure was followed. The B-spline is defined only on the top half of the plate then simmetry conditions are applied in order to minimize the time needed for the single analysis. For a VAT composite, in the most general case, which is the case of this work, all of the three independent polar parameters can vary over the structure [9]. In the mathematical framework of B-spline surfaces, the variation of the laminate anisotropic

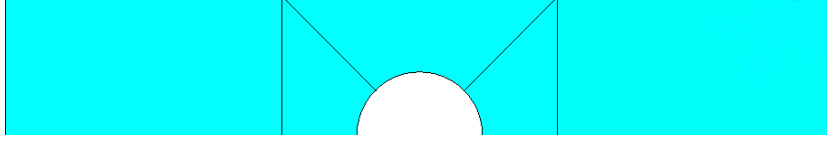


Figure 5.7: Top half plate

modulus R_{0K}^{A*} and R_1^{A*} and of the polar angle Φ_1^{A*} are stated as follows:

$$\begin{aligned} R_{0K}^{A*}(\xi, \gamma) &= \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} N_{i,p}(\xi) N_{j,q}(\gamma) R_{0K}^{A*(i,j)}; \\ R_1^{A*}(\xi, \gamma) &= \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} N_{i,p}(\xi) N_{j,q}(\gamma) R_1^{A*(i,j)}; \\ \Phi_1^{A*}(\xi, \gamma) &= \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} N_{i,p}(\xi) N_{j,q}(\gamma) \Phi_1^{A*(i,j)}. \end{aligned} \quad (5.4)$$

Eq. 5.4 fully represents a B-spline surface where, p and q are the degrees of the respectively B-spline basis function $N_{i,p}$ and $N_{j,q}$, ξ and γ represent the dimensionless coordinates linked to the Cartesian coordinates of the laminated plate,

$$\xi = \frac{x}{a}, \quad \gamma = \frac{y}{b} \quad (5.5)$$

where a and b are the length and width of the plate along x and y and $\Phi_1^{A*(i,j)}$ is the value of the laminate polar angle at the generic control point [9]. For sake of clarity, since the B-spline is defined only on half plate, the adimensional coordinates along y are computed by dividing the value of y only with $b/2$.

The use of this formulation to describe the mechanical design variables is convenient because presents the following advantages [9]:

- use an iso-geometric surface for describing the variation of R_{0K}^{A*} , R_1^{A*} and Φ_1^{A*} over the structure implies the no discontinuity of the three independent polar parameters over the plate;
- thanks to the B-spline representation the laminate polar parameters are determined solely on each control point of the *control net*, implying in this way a significant reduction in the number of design variable involved.

Therefore, the optimisation variables of the problem can be grouped into the vector:

$$\mathbf{x} = \left\{ R_{0K}^{A*(0,0)}, \dots, R_{0K}^{A*(n_u, m_v)}, R_1^{A*(0,0)}, \dots, R_1^{A*(n_u, m_v)}, \Phi_1^{A*(0,0)}, \dots, \Phi_1^{A*(n_u, m_v)} \right\}. \quad (5.6)$$

The total number of the design variables is hence equal to $3 \times (n_u + 1) \times (m_u + 1)$. BIANCA returns the values of R_{0K}^{A*} , R_1^{A*} and Φ_1^{A*} for each chromosome, i.e. the control

point, at every generation, along with the objective function value; regards this work, a **FIXED GENERATION** stop criteria is settled at 150 generations and the starting point for **fmincon** optimization consists of the values of the polar parameters for each chromosome at the last generation.

5.6.3 Plate subjected to a Tensile Force

Concerning the VAT plate, the traction analysis is performed twice, changing the number of control points and consequently, the parameters defining the B-spline surface, as well as the number of constraints. The two configurations input parameters, along with the results the gradient descent algorithm performed by **fmincon** and the distribution of the optimisation variables aon the structure, are given in the following sections in complete and extended form.

5.6.3.1 First Configuration

In the first configuration, the parameters defining the B-spline surface are set as follows:

- $n_u = 7$ and $n_v = 2$ (hence 8 control points along the x -direction and 3 control points along the y -direction, for a total of 24 for each half of the plate),
- $p = q = 2$ (degrees of the blending functions along each direction).

Moreover, the B-spline is defined over the following uniform knot-vectors:

$$\begin{aligned} U &= \{0 \ 0 \ 0 \ 0.1667 \ 0.333 \ 0.50 \ 0.6667 \ 0.8333 \ 1 \ 1 \ 1\} \\ V &= \{0 \ 0 \ 0 \ 1 \ 1 \ 1\} \end{aligned} \quad (5.7)$$

Since every polar parameters is represented by means of a B-spline surface, it's suffice to list the values of the three polar parameters R_{0K}^{A*} , R_1^{A*} and Φ_1^{A*} , in each control points to define the B-spline surface.

The first step of the optimization strategy, performed by the Genetic Algorithm BIANCA, returns the values of the $\mathbf{x}$ vector for two different populations, identifying all feasible initial points since every triplet, which is associated with a chromosome, i.e. control point, satisfy the volume constraint 5.3.

The values indicated in Table 5.16, 5.17 and 5.18 represent the dimensional results of the second step of the optimization strategy.

n_v	n_u							
	0	1	2	3	4	5	6	7
0	-82.8813	9459.2466	-12048.7699	14511.8710	-1330.8238	-987.6821	4238.8484	11720.6625
1	11987.4720	-2054.6110	15109.5183	-1186.3902	12373.1642	16729.9236	3481.9918	17209.5304
2	-1708.1173	5316.0812	14286.3118	13098.5065	14718.3024	7461.8439	17077.8420	12269.7404

Table 5.16: Traction: first configuration. Optimum value of R_{0K}^{A*} for each control point of the B-spline surface after **fmincon** optimisation

n_v	n_u							
	0	1	2	3	4	5	6	7
0	2587.4659	11653.2840	1054.6358	9194.1354	4160.8775	7248.9277	4237.1816	12751.8749
1	4295.5383	9508.3633	12886.5374	10368.7720	11152.0656	13718.9320	8060.6601	15557.0176
2	7125.8999	14170.0946	15869.7981	11898.2344	15719.9439	12565.4741	6689.4775	9987.5886

Table 5.17: Traction: first configuration. Optimum value of R_1^{A*} for each control point of the B-spline surface after **fmincon** optimisation

n_v	n_u							
	0	1	2	3	4	5	6	7
0	7.4724	5.5286	16.7837	33.1412	−80.8743	21.0034	7.6254	31.9055
1	49.1421	46.3208	4.2582	9.6918	23.4234	−55.5737	13.2198	−37.8037
2	3.1890	−35.1636	2.3094	6.6710	−8.2783	14.4136	9.1234	7.0125

Table 5.18: Traction: first configuration. Optimum value of Φ_1^{A*} for each control point of the B-spline surface after **fmincon** optimisation

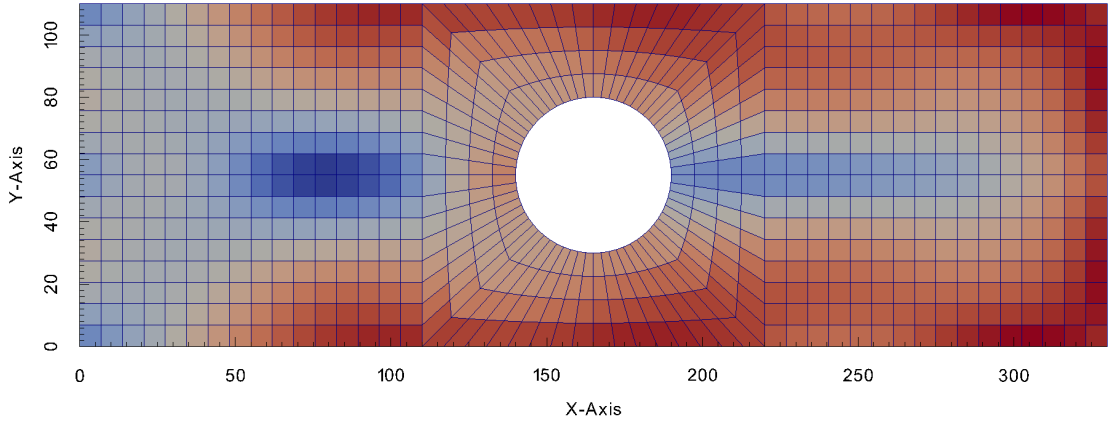


Figure 5.8: Traction: first configuration. Optimal distribution of the anisotropic modulus R_{0K}^{A*} over the VAT plate resulting from the second step of the optimisation

In order to clarify better the meanings of the results obtained, an illustration of the placement of the anisotropic polar parameters R_{0K}^{A*} , R_1^{A*} , as well of the polar angle Φ_1^{A*} is depicted in Figures 5.8, 5.9 and 5.10.

The optimal solution for the first population found at the end of the second step of the optimisation process, is characterised by a strain energy value of $0.056109 \text{ N} \cdot \text{mm}$.

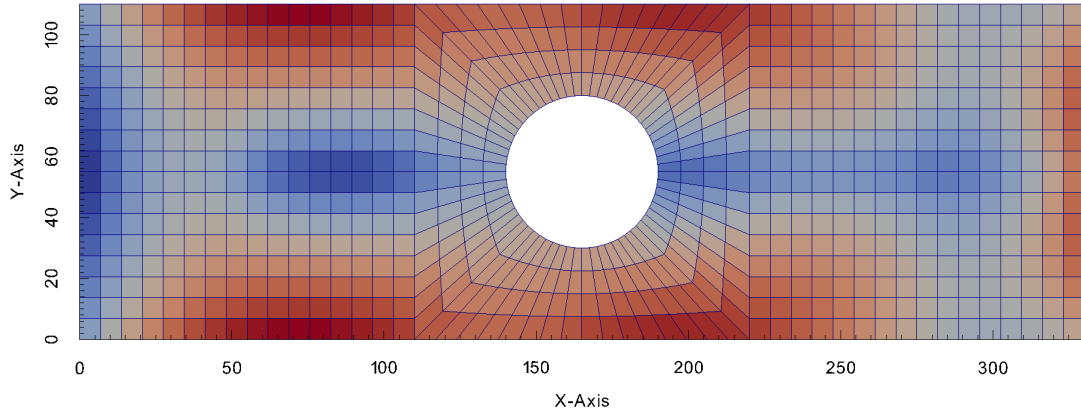


Figure 5.9: Traction: first configuration. Optimal distribution of the anisotropic modulus R_1^{A*} over the VAT plate resulting from the second step of the optimisation

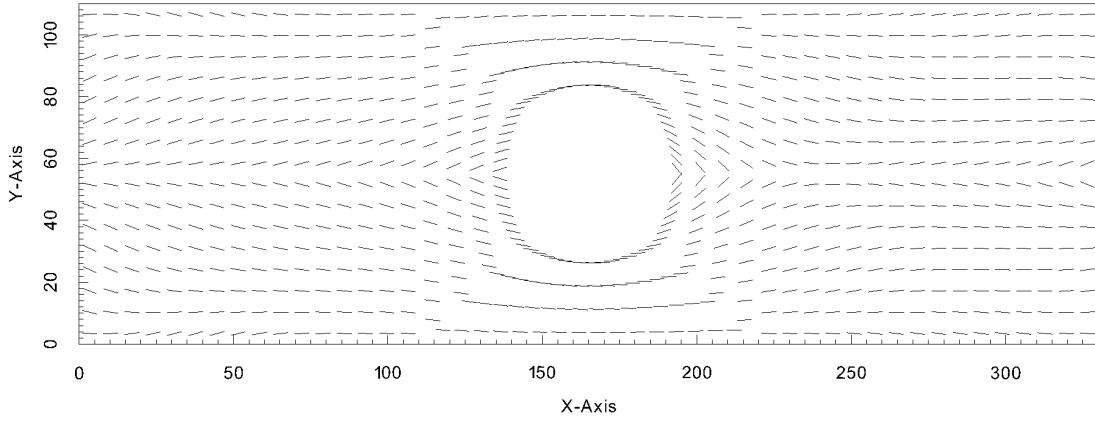


Figure 5.10: Traction: first configuration. Optimal distribution of the polar angle Φ_1^{A*} over the VAT plate resulting from the second step of the optimisation

5.6.3.2 Second Configuration

In the second configuration, the parameters defining the B-spline surface are set as follows:

- $n_u = 6$ and $n_v = 3$ (hence 7 control points along the x -direction and 4 control points along the y -direction, for a total of 28 for each half of the plate),
- $p = q = 2$ (degrees of the blending functions along each direction).

Moreover, the B-spline is defined over the following uniform knot-vectors:

$$\begin{aligned} U &= \{0 \ 0 \ 0 \ 0.2 \ 0.4 \ 0.6 \ 0.8 \ 1 \ 1 \ 1\} \\ V &= \{0 \ 0 \ 0 \ 0.5 \ 1 \ 1 \ 1\} \end{aligned} \quad (5.8)$$

The procedure followed for the second configuration analysis is the same of the one followed for the first configuration.

n_v	n_u						
	0	1	2	3	4	5	6
0	10878.9204	−8457.5450	−11916.6552	16932.3765	−15341.1925	−7246.8573	5759.4274
1	16482.6991	12781.4372	8111.6393	13092.7523	−3095.9180	13680.8097	12020.4269
2	3926.0917	17174.5105	7592.7631	14095.8966	−985.8489	12366.3503	12124.1986
3	−743.7096	11812.8835	−5655.6557	14649.3634	−1020.4395	16517.2896	−15963.8228

Table 5.19: Traction: second configuration. Optimum value of R_{0K}^{A*} for each control point of the B-spline surface after **fmincon** optimisation

n_v	n_u						
	0	1	2	3	4	5	6
0	15156.9754	5257.4071	5947.2058	2293.1305	1081.3101	4754.0379	7718.3146
1	8613.1952	6338.7173	2610.0510	8370.8274	988.0935	13945.1742	8967.4208
2	11819.8397	13497.7243	14579.0345	14933.2600	9806.3631	15865.4265	1789.7537
3	2125.3344	7848.8248	652.5147	9340.2798	9880.9349	11297.8181	2274.4779

Table 5.20: Traction: second configuration. Optimum value of R_1^{A*} for each control point of the B-spline surface after **fmincon** optimisation

n_v	n_u						
	0	1	2	3	4	5	6
0	−14.3401	−41.0850	83.8416	10.8212	54.9853	−54.1056	−21.9062
1	25.9531	4.3108	5.3666	28.0645	12.0528	28.4164	−4.1349
2	−82.7859	28.0645	−4.6627	−18.3871	−13.8123	0.0879	15.9237
3	−11.3490	21.0264	−34.2228	28.4164	−31.2317	13.6363	1.4956

Table 5.21: Traction: second configuration. Optimum value of Φ_1^{A*} for each control point of the B-spline surface after **fmincon** optimisation

As before, illustrations of the placement of the anisotropic polar parameters R_{0K}^{A*} , R_1^{A*} , as well of the polar angle Φ_1^{A*} , along the entire geometry of the plate is depicted in Figures 5.11, 5.12 and 5.13.

The optimal solution for the first population found at the end of the second step of the optimisation process, is characterised by a strain energy value of $0.0614186 \text{ N} \cdot \text{mm}$. The values of the strain energy obtained by increasing to 28 the number of control points is slightly higher than the value returned by the first analysis performed with 24 control points displaying a better evaluation of the objective function.

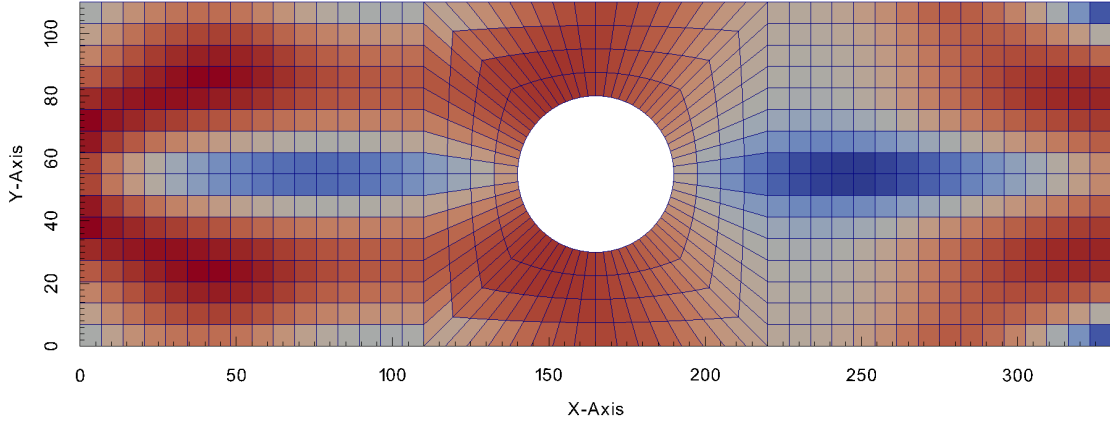


Figure 5.11: Traction: second configuration. Optimal distribution of the anisotropic modulus R_{0K}^{A*} over the VAT plate resulting from the second step of optimisation

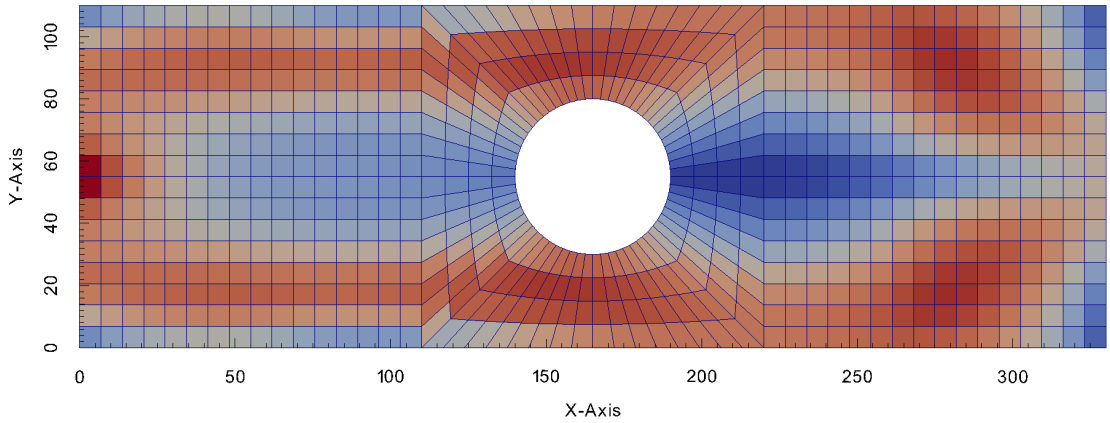


Figure 5.12: Traction: second configuration. Optimal distribution of the anisotropic modulus R_1^{A*} over the VAT plate resulting from the first-step of optimisation

5.6.4 Plate subjected to a Bending Force

As in Chapter 5.6.3, the bending analysis is performed twice, changing the number of control points and consequently, the parameters defining the B-spline surface, as well as the number of constraints. The two configurations input parameters, along with the results the gradient descent algorithm performed by `fmincon` and the distribution of the optimisation variables aon the structure, are given in the following sections in complete and extended form.

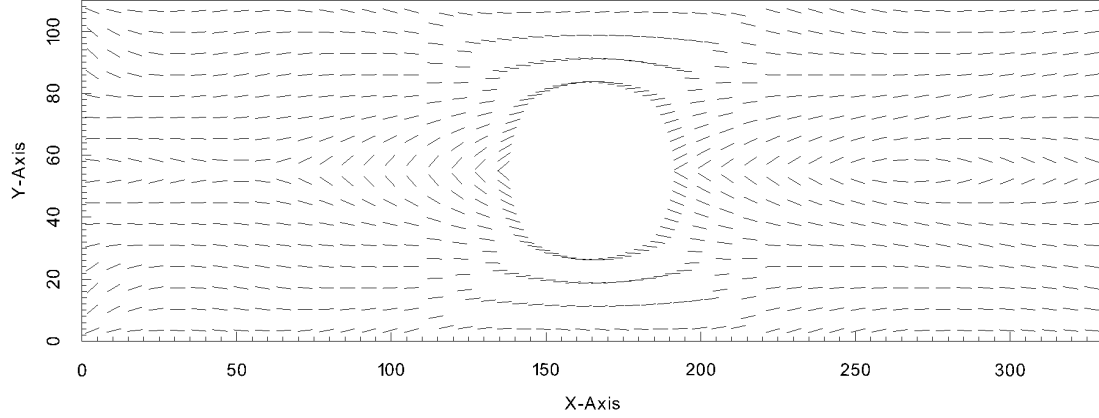


Figure 5.13: Traction: second configuration. Optimal distribution of the polar angle Φ_1^{A*} over the VAT plate resulting from the first-step of optimisation

5.6.4.1 First Configuration

In the first configuration, the parameters defining the B-spline surface are set as follows:

- $n_u = 7$ and $n_v = 2$ (hence 8 control points along the x -direction and 3 control points along the y -direction, for a total of 24 for each half of the plate),
- $p = q = 2$ (degrees of the blending functions along each direction).

Moreover, the B-spline is defined over the following uniform knot-vectors:

$$\begin{aligned} U &= \{0 \quad 0 \quad 0 \quad 0.677 \quad 0.333 \quad 0.50 \quad 0.6667 \quad 0.8333 \quad 1 \quad 1 \quad 1\} \\ V &= \{0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1\} \end{aligned} \quad (5.9)$$

Since every polar parameters is represented by means of a B-spline surface, it's suffice to list the values of the three polar parameters R_{0K}^{A*} , R_1^{A*} and Φ_1^{A*} , in each control points to define the B-spline surface. The first step of the optimization strategy, performed by the Genetic Algorithm BIANCA, returns the values of the $\mathbf{x}$ vector for two different populations, identifying all feasible initial points since every triplet, which is associated with a chromosome, i.e. control point, satisfy the volume constraint 5.3.

The values indicated in Table 5.22, 5.23 and 5.24 represent the dimensional results of the second step of the optimization strategy.

n_v	n_u							
	0	1	2	3	4	5	6	7
0	12972.0574	-2127.2716	16549.0097	5019.9210	2695.2478	5406.6499	15814.9351	17654.3045
1	389.6584	-3955.6844	13593.0286	15316.8171	-4117.9613	-2496.7336	1462.8800	-11503.4565
2	3088.7655	2205.9695	16435.5914	-1974.2715	184.1291	-1182.9223	455.4691	9557.2595

Table 5.22: Bending: first configuration. Optimum value of R_{0K}^{A*} for each control point of the B-spline surface after `fmincon` optimisation

n_v	n_u							
	0	1	2	3	4	5	6	7
0	9953.0294	8449.1553	15583.4873	79.5866	8165.4646	1337.0536	3167.5468	12690.2755
1	4860.8113	6673.3288	7205.3574	13367.8527	8192.9592	3213.0362	4445.2789	1289.2566
2	2941.8453	10696.4545	15759.6030	11348.7814	11653.5856	4193.6337	8282.2873	1337.0536

Table 5.23: Bending: first configuration. Optimum value of R_1^{A*} for each control point of the B-spline surface after **fmincon** optimisation

n_v	n_u							
	0	1	2	3	4	5	6	7
0	-18.8393	62.5484	-2.4861	59.9657	-5.2335	-18.5676	-9.2837	55.7838
1	-23.1512	-22.9294	-39.8177	15,0004	-27,9244	54,4931	45,5400	49.8259
2	-23.2215	24.4660	26.9365	-0,0582	-15.5127	-6.1302	60.3542	4.61573

Table 5.24: Bending: first configuration. Optimum value of Φ_1^{A*} for each control point of the B-spline surface after **fmincon** optimisation

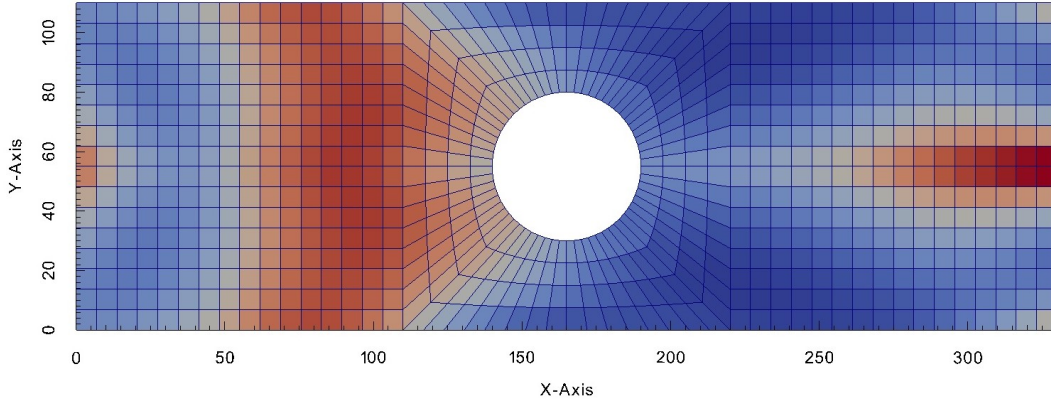


Figure 5.14: Bending: first configuration. Optimal distribution of the anisotropic modulus R_{0K}^{A*} over the VAT plate resulting from the second step of the optimisation

In order to clarify better the meanings of the results obtained, an illustration of the placement of the anisotropic polar parameters R_{0K}^{A*} , R_1^{A*} , as well of the polar angle Φ_1^{A*} is depicted in Figures 5.14, 5.15 and 5.16.

The optimal solution for the first population found at the end of the second step of the optimisation process, is characterised by a strain energy value of $653.934 \text{ N} \cdot \text{mm}$. As done for the plate subjected to a tensile stress, even in the bending study case, two different configuration have been take into account, of whom the results of the second one are given in the next chapter.

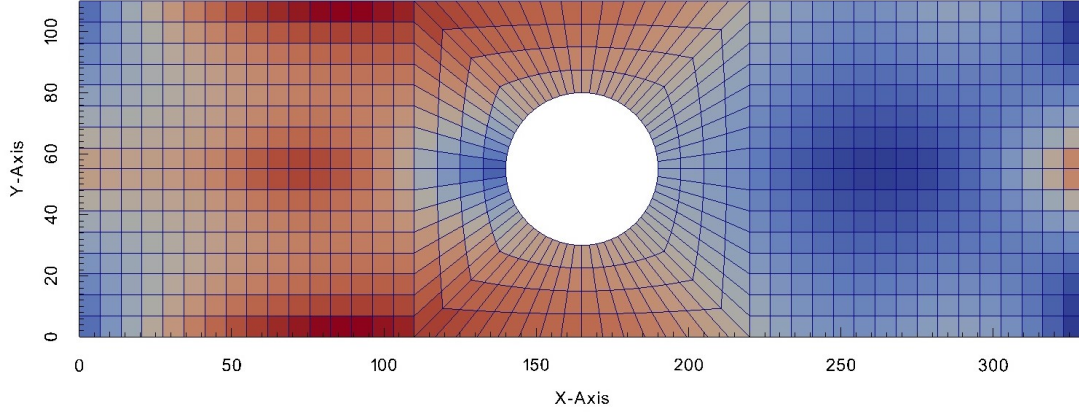


Figure 5.15: Bending: first configuration. Optimal distribution of the anisotropic modulus R_1^{A*} over the VAT plate resulting from the second step of the optimisation

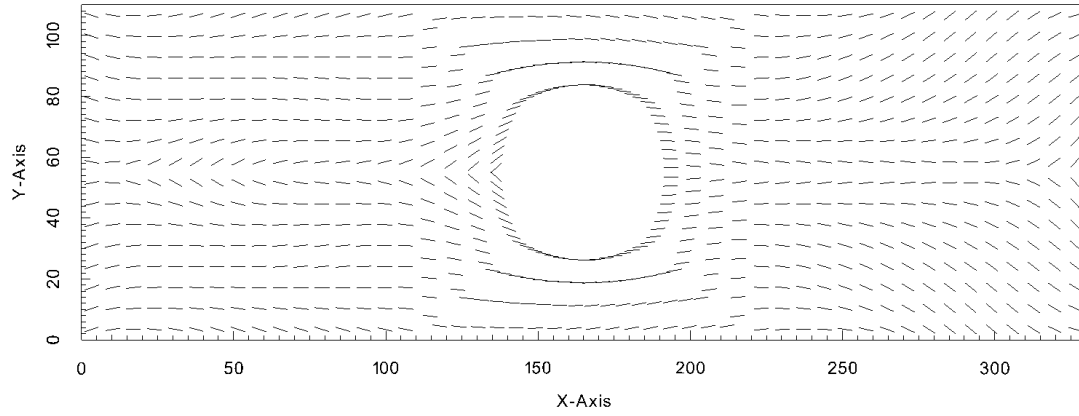


Figure 5.16: Bending: first configuration. Optimal distribution of the polar angle Φ_1^{A*} over the VAT plate resulting from the second step of the optimisation

5.6.4.2 Second Configuration

In the second configuration, the parameters defining the B-spline surface are set as follows:

- $n_u = 6$ and $n_v = 3$ (hence 7 control points along the x -direction and 4 control points along the y -direction, for a total of 28 for each half of the plate),
- $p = q = 2$ (degrees of the blending functions along each direction).

Moreover, the B-spline is defined over the following uniform knot-vectors:

$$\begin{aligned} U &= \{0 \quad 0 \quad 0 \quad 0.2 \quad 0.4 \quad 0.6 \quad 0.8 \quad 1 \quad 1 \quad 1\} \\ V &= \{0 \quad 0 \quad 0 \quad 0.5 \quad 1 \quad 1 \quad 1\} \end{aligned} \quad (5.10)$$

The procedure followed for the second configuration analysis is the same of the one followed for the first configuration.

n_v	n_u						
	0	1	2	3	4	5	6
0	-6713.3610	12360.6506	14657.7470	17121.7627	2695.5956	2679.7645	9994.8800
1	6861.8436	14952.7226	15184.2528	8622.2344	8220.1923	15561.7650	5023.5206
2	-13442.2537	12045.4871	10121.2324	15225.3567	13416.6454	-12126.5003	-10915.2060
3	1781.5792	-4547.2777	-2886.8170	17041.5341	-10876.1203	5172.9097	-4382.7633

Table 5.25: Bending: second configuration. Optimum value of R_{0K}^{A*} for each control point of the B-spline surface after **fmincon** optimisation

n_v	n_u						
	0	1	2	3	4	5	6
0	9033.1380	10593.4518	3436.8057	2554.1173	9409.8822	8771.0870	12083.0360
1	7234.1979	15840.9182	11627.8183	8065.9977	13153.6124	14485.8082	14025.7718
2	544.6127	12085.3871	8090.0403	8376.0299	2385.1069	563.4093	8062.2718
3	751.2131	4412.2929	7192.4227	14800.4772	7594.8972	15126.4826	939.0169

Table 5.26: Bending: second configuration. Optimum value of R_1^{A*} for each control point of the B-spline surface after **fmincon** optimisation

n_v	n_u						
	0	1	2	3	4	5	6
0	-13.3907	-0.2389	-11.2307	51.9069	6.1424	27.2623	72.6980
1	7.7857	-28.7606	-4.6054	-5.5153	71.3590	23.9192	9.3942
2	32.2912	35.1897	0.2601	-5.2761	-87.7045	57.1067	-39.8523
3	67.0318	-79.9083	15.9933	-21.7601	3.2316	84.5878	-52.6019

Table 5.27: Bending: second configuration. Optimum value of Φ_1^{A*} for each control point of the B-spline surface after **fmincon** optimisation

As before, illustrations of the placement of the anisotropic polar parameters R_{0K}^{A*} , R_1^{A*} , as well of the polar angle Φ_1^{A*} , along the entire geometry of the plate is depicted in Figures 5.11, 5.12 and 5.13.

The optimal solution for the first population found at the end of the second step of the optimisation process, is characterised by a strain energy value of $650.436 \text{ N} \cdot \text{mm}$. The values of the strain energy obtained by increasing to 28 the number of control points is slightly higher than the value returned by the first analysis performed with 24 control points displaying a better evaluation of the objective function. As summary, the results obtained are listed in Table 5.28.

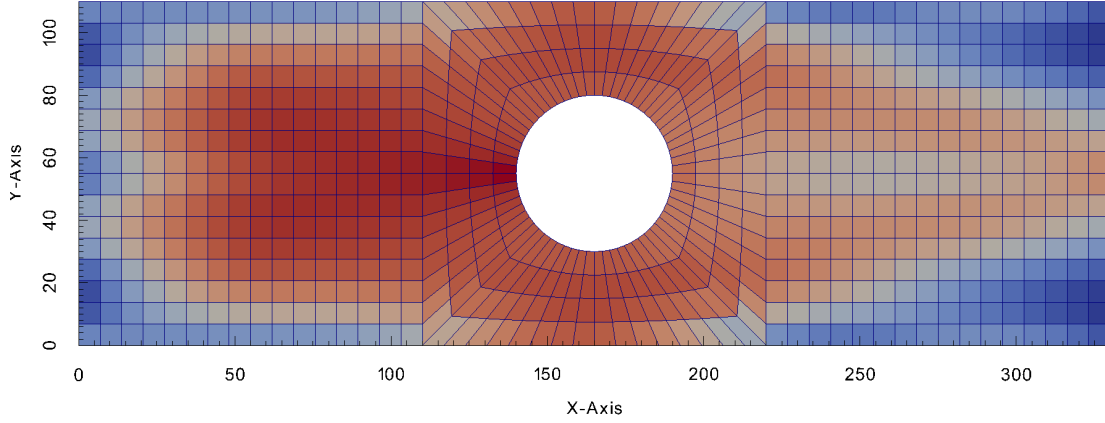


Figure 5.17: Bending: second configuration. Optimal distribution of the anisotropic modulus R_{0K}^{A*} over the VAT plate resulting from the second step of optimisation

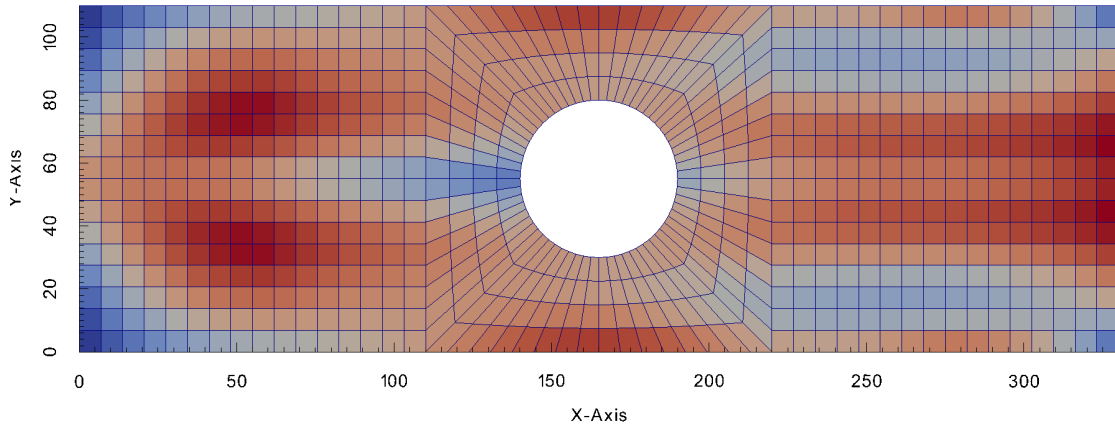


Figure 5.18: Bending: second configuration. Optimal distribution of the anisotropic modulus R_1^{A*} over the VAT plate resulting from the first-step of optimisation

Study Case	Control Points	S.E.(N*mm)
Traction	24	0.05611
Traction	28	0.06142
Bending	24	653.934
Bending	28	650.436

Table 5.28: Summary of the strain energy values

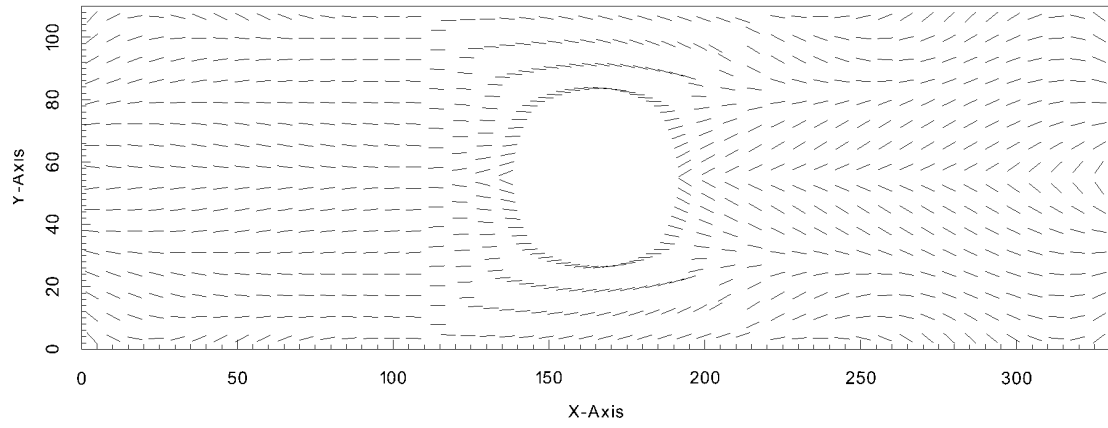


Figure 5.19: Bending: second configuration. Optimal distribution of the polar angle Φ_1^{A*} over the VAT plate resulting from the first-step of optimisation

Chapter 6

Conclusions

6.1 Final Considerations

In this thesis, the analytical formulae of the derivatives of the strain energy as well as of the feasibility constraints for VAT composites to be integrated into MS2L optimisation strategy (an effective and general method for the design of VAT composites) have been derived. [10].

Regarding this work, the optimisation strategy has been applied to solve the first-level problem (laminate macroscopic scale) focusing on the minimization of the strain energy of a VAT multilayered plate subjected to both a volume constraint (on the overall number of layers) and feasibility constraints on the laminate polar parameters. The proposed design process is not submitted to restrictions: any parameter characterising the VAT composite (at each scale) is an optimisation variable. This allows the designer to look for a true global minimum, hard to be obtained otherwise.

The framework of the two-level approach used relies on two crucial assumption:

- the use of *high order theories*, to take into account the influence of the shear stiffness on the VAT structure;
- the use of *B-spline hypersurfaces*, to describe the polar parameters point-wise distribution over the laminate structure.

This last point leads to some important advantages for the resolution of the related optimisation problem. Indeed, B-spline hypersurfaces leads to a considerable reduction in the number of design variables (the polar parameters are defined solely in each point of the control network of the B-spline hypersurface). Secondly, thanks to the strong *convex-hull* property of the B-spline blending functions the optimisation constraints of the problem can be imposed only on the control points: if they are met on such points they are automatically fulfilled over the whole domain [10]. The use of the B-spline hypersurface does not affect the accuracy of the results because, thanks to the *local support property* of B-spline blending functions [15], each control point affects the shape of the B-spline hypersurface within its influence region, i.e. the control point local support, which is a sort of neighbourhood defined for each control point.

In this study all the laminate independent polar parameters (for a quasi-homogeneous fully orthotropic laminate), i.e. R_{0K}^{A*} , R_1^{A*} and Φ_1^{A*} vary over the structure, which allows to obtain the most general combination returning the global minimum of the objective function. The original and creative part of the optimisation research and study carried out in this work relies on the use of the energy derivatives with respect to all the three polar parameters that here represents the optimisation variables. Taking into account all the values of the derivatives allows obtaining a more efficient and complete gradient descent optimisation, which would be difficult to obtain in other ways.

It has to be said that, since only simple geometric and mechanical constraints are taken into account, regardless of manufacturability possibilities, the results proposed may not represent a feasible configuration for the actual fabrication technology.

Concerning the future perspectives of this thesis, more aspects are needed to be studied and investigated. First of all, a manufacturability constraint must be taken into account, like the minimum radius achievable by the actual AFP technology, the variation of the volume fiber, the possible presence of tow gaps and overlaps... in order to achieve a feasible and realistic result.

Research studies are currently ongoing on these aspects.

Appendix A

Algorithms

The following algorithms were described in Chapter 2 and then implemented on MATLAB in order to be tested

A.1 Unconstrained Algorithms

A.1.1 Line Search Algorithms

A.1.1.1 Step Length Algorithms

Algorithm 1 Armijo Backtracking Method

Set: $\alpha_{in} > 0, \rho \in (0,1), c_1 \in (0,1)$
 $\alpha \leftarrow \alpha_{in}$
 Positive Direction
 while $f(x_k + \alpha p_k) > f(x_k) + c_1 \alpha \nabla f_k^T p_k$ **do**
 $\alpha_1 \leftarrow \rho \alpha$
Set: $f_{k1} \leftarrow f(x_k + \alpha_1 p_k) - f(x_k) - c_1 \alpha_1 \nabla f_k^T p_k$
 end while
 Negative Direction
 while $f(x_k - \alpha p_k) > f(x_k) - c_1 \alpha \nabla f_k^T p_k$ **do**
 $\alpha_2 \leftarrow -\rho \alpha$
Set: $f_{k2} \leftarrow f(x_k + \alpha_2 p_k) - f(x_k) - c_1 \alpha_2 \nabla f_k^T p_k$
 end while
 if $f_{k1} < f_{k2}$ **then**
 $\alpha \leftarrow \alpha_1$
 else
 $\alpha \leftarrow \alpha_2$
 end if

Algorithm 2 Golden Section Method

Set: $\varphi = (1 + \sqrt{5})/2, I = [a, b], tol$
Compute: $f(a), f(b)$
Set: $c = b - (b - a)/\varphi$
 $d = a + (b - a)/\varphi$
while $\| (c - d) \| > tol$ **do**
 if $f(c) < f(d)$ **then**
 $b \leftarrow d$
 else
 $a \leftarrow c$
 end if
end while

Algorithm 3 Cubic Interpolation Method

Set: $x_0, \alpha_1, \alpha_2, f(x_0), \nabla f(x_0), f(\alpha_1), f(\alpha_2)$
 $\rho \leftarrow \frac{1}{\alpha_1^2 \alpha_2^2 (\alpha_1 - \alpha_2)}$
 $[A] \leftarrow \begin{bmatrix} \alpha_2^2 & -\alpha_1^2 \\ -\alpha_2^3 & \alpha_1^3 \end{bmatrix}$
 $[B] \leftarrow \begin{bmatrix} f(\alpha_1) - f(x_0) - \alpha_1 \nabla f(x_0) \\ f(\alpha_2) - f(x_0) - \alpha_2 \nabla f(x_0) \end{bmatrix}$
Solve: $\begin{pmatrix} a \\ b \end{pmatrix} = \rho AB$
if $a = 0$ **then**
 $\alpha \leftarrow -\frac{\nabla f(x_0)}{2b}$
else if $d = b^2 - 3a \nabla f(x_0)$ **then**
 $\alpha \leftarrow \frac{-b + \sqrt{d}}{3a}$
end if
if $\alpha < a$ **then**
 return α
else
 if $\alpha > b$ **then**
 return b
 else
 return a
 end if
end if

A.1.1.2 Descent Algorithms

Algorithm 4 Gradient Descent Method

Set: $x_0, \epsilon, kmax$ **Set:** $x_k \leftarrow x_0, k \rightarrow 0$ **Compute:** $\nabla f(x_k)$ **while** $\|\nabla f(x_k)\| > \epsilon \ \& \ k < kmax$ **do**

$$p_k = -\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}$$

Compute: α_k {with the chosen step length method}

$$x_{k+1} \leftarrow x_k + \alpha_k p_k$$

$$x_k \leftarrow x_{k+1}$$

$$k \leftarrow k + 1$$

end while

Algorithm 5 Newton Method

Set: $x_0, \epsilon, kmax$ **Set:** $x_k \leftarrow x_0, k \rightarrow 0$ **Compute:** $\nabla f(x_k), \nabla^2 f(x_k)$ **while** $\|\nabla f(x_k)\| > \epsilon \ \& \ k < kmax$ **do**

$$p_k = -\frac{\nabla^2 f(x_k)^{-1} \nabla f(x_k)}{\|\nabla f(x_k)\|}$$

Compute: α_k

$$x_{k+1} \leftarrow x_k + \alpha_k p_k$$

$$x_k \leftarrow x_{k+1}$$

$$k \leftarrow k + 1$$

end while

Algorithm 6 Quasi Newton Method

Set: $x_0, \epsilon, kmax$ **Set:** $x_k \leftarrow x_0, k \rightarrow 0, \nabla^2 f(x_k) \rightarrow I$ **Compute:** $\nabla f(x_k)$ **while** $\|\nabla f(x_k)\| > \epsilon \ \& \ k < kmax$ **do**

$$p_k = -\frac{\nabla^2 f(x_k)^{-1} \nabla f(x_k)}{\|\nabla f(x_k)\|}$$

Compute: α_k {with the chosen step length method}

$$x_{k+1} \leftarrow x_k + \alpha_k p_k$$

Compute: $\nabla^2 f(x_k)$ {with $(.)$ $(.)$ }

$$x_k \leftarrow x_{k+1}$$

$$k \leftarrow k + 1$$

end while

Algorithm 7 Conjugate Gradient Method

Set: x_0
Set: $r_0 \leftarrow Ax_0 - b, p_0 \leftarrow -r_0, k \rightarrow 0$,
while $r_k \neq 0$ **do**
 $\alpha_k \leftarrow \frac{r_k^T r_k}{p_k^T A p_k}$
 $x_{k+1} \leftarrow x_k + \alpha_k p_k$
 $r_{k+1} \leftarrow r_k + \alpha_k p_k$
 $\beta_{k+1} \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
 $p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k$
 $k \leftarrow k + 1$
end while

Algorithm 8 Preconjugate Gradient Method

Set: x_0, M {preconditioner}
Set: $r_0 \leftarrow Ax_0 - b$
Solve: $My_0 = r_0 \leftarrow f$ *or* y_0
Set: $p_0 \leftarrow -r_0, k \rightarrow 0$
while $r_k \neq 0$ **do**
 $\alpha_k \leftarrow \frac{r_k^T y_k}{p_k^T A p_k}$
 $x_{k+1} \leftarrow x_k + \alpha_k p_k$
 $r_{k+1} \leftarrow r_k + \alpha_k p_k$
 $M_{k+1} \leftarrow r_{k+1}$
 $\beta_{k+1} \leftarrow \frac{r_{k+1}^T y_{k+1}}{r_k^T r_k}$
 $p_{k+1} \leftarrow -y_{k+1} + \beta_{k+1} p_k$
 $k \leftarrow k + 1$
end while

Algorithm 9 Fletcher-Reeves Method

Set: x_0
Compute: $f_0 = f(x_0), \nabla f_0 = \nabla f(x_0)$
Set: $p_0 = -\nabla f_0, k \rightarrow 0$
while $\nabla f(x_k) \neq 0$ **do**
Compute: α_k {with the chosen step length method}
Compute: $\nabla f(x_{k+1})$
 $\beta_{k+1}^{FR} \leftarrow \frac{\nabla f(x_{k+1})^T \nabla f(x_{k+1})}{\nabla f(x_k)^T \nabla f(x_k)}$
 $p_{k+1} \leftarrow -\nabla f(x_k) + \beta_{k+1}^{FR} p_k$
 $k \leftarrow k + 1$
end while

Algorithm 10 Polak-Ribière Method

Set: x_0 **Compute:** $f_0 = f(x_0), \nabla f_0 = \nabla f(x_0)$ **Set:** $p_0 = -\nabla f_0, k \rightarrow 0$ **while** $\nabla f(x_k) \neq 0$ **do****Compute:** α_k {with the chosen step length method}**Compute:** $\nabla f(x_{k+1})$

$$\beta_{k+1}^{PR} \leftarrow \frac{\nabla f(x_{k+1})^T (\nabla f(x_{k+1}) - \nabla f(x_k))}{\nabla f(x_k)^T \nabla f(x_k)}$$

$$p_{k+1} \leftarrow -\nabla f(x_k) + \beta_{k+1}^{PR} p_k$$

$$k \leftarrow k + 1$$

end while

A.1.2 Trust Region Algorithm

Algorithm 11 Trust Region Method with BSFG

Set: $\Delta_{max}, \Delta_{in} \in (0, \Delta_{max}), \mu \in \left[0, \frac{1}{4}\right)$
Set: $k \rightarrow 0, B \rightarrow I$
while $\|\nabla f(x_k)\| > \epsilon$ & $k < k_{max}$ **do**
 Compute: p_k {by solving approximately $(.)$ }
 Compute: ρ_k {from $(.)$ }
 if $\rho_k < \frac{1}{4}$ **then**
 $\Delta_{k+1} = \frac{1}{4} \|p_k\|$
 else if $\rho_k > \frac{3}{4}$ & $\|p_k\| = \Delta_k$ **then**
 $\Delta_{k+1} = \min(2\Delta_k, \Delta_{k+1})$
 else
 $\Delta_{k+1} = \Delta_k$
 end if
 if $\rho_k > \mu$ **then**
 $x_{k+1} \leftarrow x_k + p_k$
 else
 $x_{k+1} \leftarrow x_k$
 end if
 Compute: $\nabla^2 f(x_k)$ {with $(.)$ $(.)$ }
 $k \leftarrow k + 1$
end while

A.1.2.1 Trust Region Subproblem Algorithms

Algorithm 12 Exact Trust Region Method

Set: $\Delta, \nabla f(x_k), B$
Set: $r = B^{-1} \nabla f(x_k)$
 Compute: $\lambda^l = |2(\max(\text{eig}(B)))|$
 for $k = 1 : 3$ **do**
 Compute: $B + \lambda I = R^T R$
 Solve: $R^T R P_l = -\nabla f(x_k)$
 Solve: $R^T q_l = p_l$
 Set: $\lambda^{l+1} \leftarrow \lambda^l + \left(\frac{\|p_l\|}{\|q_l\|}\right)^2 \left(\frac{\|p_l\| - \Delta}{\Delta}\right)$
 end for

Algorithm 13 Cauchy Point Method

Set: $\Delta, \nabla f(x_k), B$
if $\nabla f(x_k)^T B_k \nabla f(x_k) \leq 0$ **then**
 $r \leftarrow -\Delta \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}$
else
 $\alpha \leftarrow \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T B_k \nabla f(x_k)}$
end if
if $\|\alpha \nabla f(x_k)\| \leq \Delta$ **then**
 $r \leftarrow -\alpha \nabla f(x_k)$
else
 $r \leftarrow -\Delta \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}$
end if

Algorithm 14 Dogleg Method

Set: $\Delta, \nabla f(x_k), B$
Set: $s_n \leftarrow -B^{-1} \nabla f(x_k)$
 $s_g \leftarrow -\nabla f(x_k) \frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^T B_k \nabla f(x_k)}$
if $\|s_g\| > \Delta$ **then**
 $r \leftarrow \Delta \frac{s_g}{\|s_g\|}$
else if $\|s_g\| < \Delta$ **then**
 $r \leftarrow s_n$
else
 $a \leftarrow \|s_g\|^2$
 $b \leftarrow \|s_n\|^2$
 $c \leftarrow \|s_g - s_n\|^2$
 $\alpha \leftarrow \frac{\frac{a+b-c}{2}}{b-d+\sqrt{d^2-ab+\Delta^2c}}$
 $r \leftarrow \alpha s_g + (1 - \alpha) s_n$
end if
return r

Algorithm 15 Double Dogleg Method

Set: $\Delta, \nabla f(x_k), B$
Set: $s_n \leftarrow -B^{-1} \nabla f(x_k)$
 $s_g \leftarrow -\nabla f(x_k) \frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^T B_k \nabla f(x_k)}$
 $\gamma \leftarrow \frac{\|s_g\|^2}{s_g^T s_n}$
if $\|s_g\| > \Delta$ **then**
 $r \leftarrow \Delta \frac{s_g}{\|s_g\|}$
else if $\gamma \|s_n\| \geq \Delta$ **then**
 $a \leftarrow \gamma^2 \|s_n\|^2 - \|s_g\|^2$
 $b \leftarrow \Delta^2 - \|s_g\|^2$
 $\alpha \leftarrow \frac{a-b}{a+\sqrt{a^2+b^2}}$
 $r \leftarrow \alpha s_g + (1-\alpha) s_n$
else if $\|s_n\| \geq \Delta$ **then**
 $r \leftarrow \frac{\Delta s_n}{\|s_n\|}$
else
 $r \leftarrow \|s_n\|$
end if
return r

Algorithm 16 Penalty Method

Set: $x_0, \varepsilon, k \rightarrow 0, err$
while $err > \varepsilon \ \& \ k < kmax$ **do**
 Compute: P_k with {4.15}
 Compute: x_k {with the chosen descent method}
Set: $err = \nabla P(x_k)$
 $k \leftarrow k + 1$
end while

Algorithm 17 Lagrangian and Augmented Lagrangian Method

Set: $x_0, \varepsilon, k \rightarrow 0, err$
while $err > \varepsilon \ \& \ k < kmax$ **do**
 Compute: L_k with {4.16} or {4.19}
 Compute: x_k {with the chosen descent method}
Set: $err = \nabla L(x_k)$
 $k \leftarrow k + 1$
end while

Appendix B

Homogenised Matrices and Derivatives

In this section the polar parameters of the homogenised matrices are reported and expressed as functions of the polar parameters of the lamina reduced stiffness matrices and of the geometrical properties i.e layer orientation and position. An extended discussion of this topic can be found in [7].

The polar parameter R_{0K}^{A*} is introduced using the relation $R_{0K}^{A*} = (-1)^k R_0^{A*}$ where k is the number of the ply. For reason of simplification will be written $c4\varphi$ and $c2\varphi$ instead of $\cos(4\Phi_1^{A*})$ and $\cos(2\Phi_1^{A*})$ as well as $s4\varphi$ and $s2\varphi$ instead of $\sin(4\Phi_1^{A*})$ and $\sin(2\Phi_1^{A*})$.

Moreover, the extended form of the homogenised matrices and the respective derivatives with respect to $[A^*]$ polar parameters R_{0K}^{A*} , R_1^{A*} and Φ_1^{A*} are given below.

B.1 Homogenised Matrices

Homogenised Matrix $[A^*]$

$$\begin{aligned} T_0^{A*} &= T_0, \\ T_1^{A*} &= T_1, \\ R_0^{A*} e^{i4\Phi_0^{A*}} &= \frac{1}{n} R_0 e^{i4\Phi_0} \sum_{k=1}^n e^{i4\delta_k}, \\ R_1^{A*} e^{i2\Phi_1^{A*}} &= \frac{1}{n} R_1 e^{i2\Phi_1} \sum_{k=1}^n e^{i2\delta_k}. \end{aligned} \tag{B.1}$$

$$[A^*] = \begin{bmatrix} T_0^{A*} + 2T_1^{A*} + R_{0K}^{A*} c4\varphi + 4R_1^{A*} c2\varphi & -T_0^{A*} + 2T_1^{A*} - R_{0K}^{A*} c4\varphi & R_{0K}^{A*} s4\varphi + 2R_1^{A*} s2\varphi \\ -T_0^{A*} + 2T_1^{A*} - R_{0K}^{A*} c4\varphi & T_0^{A*} + 2T_1^{A*} + R_{0K}^{A*} c4\varphi - 4R_1^{A*} c2\varphi & -R_{0K}^{A*} s4\varphi + 2R_1^{A*} s2\varphi \\ R_{0K}^{A*} s4\varphi + 2R_1^{A*} s2\varphi & -R_{0K}^{A*} s4\varphi + 2R_1^{A*} s2\varphi & T_0^{A*} - R_{0K}^{A*} c4\varphi \end{bmatrix} \tag{B.2}$$

Homogenised Matrix $[B^*]$

$$\begin{aligned}
 T_0^{B^*} &= 0, \\
 T_1^{B^*} &= 0, \\
 R_0^{B^*} e^{i4\Phi_0^{B^*}} &= \frac{1}{n^2} R_0 e^{i4\Phi_0} \sum_{k=1}^n b_k e^{i4\delta_k}, \\
 R_1^{B^*} e^{i2\Phi_1^{B^*}} &= \frac{1}{n^2} R_1 e^{i2\Phi_1} \sum_{k=1}^n b_k e^{i2\delta_k}.
 \end{aligned} \tag{B.3}$$

The expression of $[B^*]$ is not provided because $[B^*] = [0]$.

Homogenised Matrix $[D^*]$

$$\begin{aligned}
 T_0^{D^*} &= T_0, \\
 T_1^{D^*} &= T_1, \\
 R_0^{D^*} e^{i4\Phi_0^{D^*}} &= \frac{1}{n^3} R_0 e^{i4\Phi_0} \sum_{k=1}^n d_k e^{i4\delta_k}, \\
 R_1^{D^*} e^{i2\Phi_1^{D^*}} &= \frac{1}{n^3} R_1 e^{i2\Phi_1} \sum_{k=1}^n d_k e^{i2\delta_k}.
 \end{aligned} \tag{B.4}$$

The expression of $[D^*]$ is not provided because $[D^*] = [A^*]$.

Homogenised Matrix $[H^*]$

$$\begin{aligned}
 T_{H^*} &= \begin{cases} T & \text{(basic),} \\ 2T & \text{(modified),} \end{cases} \\
 R_{H^*} e^{i2\Phi_{H^*}} &= \begin{cases} R_1^{A^*} \frac{R}{R_1} e^{i2(\Phi_1^{A^*} + \Phi - \Phi_1)} \\ \frac{R}{R_1} e^{i2(\Phi - \Phi_1)} (3R_1^{A^*} e^{i2\Phi_1^{A^*}} - R_1^{D^*} e^{i2\Phi_1^{D^*}}) \end{cases}
 \end{aligned} \tag{B.5}$$

$$[H^*] = \begin{bmatrix} T + R_1^{A^*} \frac{R}{R_1} [\cos 2(\Phi + \Phi_1 - \Phi_1^{A^*})] & R_1^{A^*} \frac{R}{R_1} [\sin 2(\Phi + \Phi_1 - \Phi_1^{A^*})] \\ R_1^{A^*} \frac{R}{R_1} [\sin 2(\Phi + \Phi_1 - \Phi_1^{A^*})] & T - R_1^{A^*} \frac{R}{R_1} [\cos 2(\Phi + \Phi_1 - \Phi_1^{A^*})] \end{bmatrix} \tag{B.6}$$

The homogenised out-of-plane shear stiffness matrix is here straight expressed as functions of the polar parameters of $[A^*]$ while the correct and complete discussion can be found in the corrigendum of [6].

B.2 Homogenised Matrices Derivatives

B.2.1 Homogenised Matrix $[A^*]$ Derivatives

Derivative with respect to $R_{0k}^{A^*}$

$$\left[\frac{\partial[A^*]}{R_{0K}^{A^*}} \right] = \begin{bmatrix} c4\varphi & -c4\varphi & s4\varphi \\ -c4\varphi & c4\varphi & -s4\varphi \\ s4\varphi & -s4\varphi & -c4\varphi \end{bmatrix} \quad (\text{B.7})$$

Derivative with respect to $R_1^{A^*}$

$$\left[\frac{\partial[A^*]}{R_1^{A^*}} \right] = \begin{bmatrix} 4c2\varphi & 0 & 2s2\varphi \\ 0 & -4c2\varphi & 2s2\varphi \\ 2s2\varphi & 2s2\varphi & 0 \end{bmatrix} \quad (\text{B.8})$$

Derivative with respect to $\Phi_1^{A^*}$

$$\left[\frac{\partial[A^*]}{\Phi_1^{A^*}} \right] = \begin{bmatrix} -4R_{0K}^{A^*} s4\varphi - 8R_1^{A^*} s2\varphi & 4R_{0K}^{A^*} s4\varphi & 4R_{0K}^{A^*} c4\varphi + 4R_1^{A^*} c2\varphi \\ 4R_{0K}^{A^*} s4\varphi & -4R_{0K}^{A^*} s4\varphi + 8R_1^{A^*} s2\varphi & -4R_{0K}^{A^*} c4\varphi + 4R_1^{A^*} c2\varphi \\ 4R_{0K}^{A^*} c4\varphi + 4R_1^{A^*} c2\varphi & -4R_{0K}^{A^*} c4\varphi + 4R_1^{A^*} c2\varphi & 4R_{0K}^{A^*} s4\varphi \end{bmatrix} \quad (\text{B.9})$$

B.2.2 Homogenised Matrix $[D^*]$ Derivatives

Due to the approximation $[A^*] = [D^*]$ the expression of the derivatives of $[D^*]$ are the same as the ones of $[A^*]$ and therefore not re-written.

B.2.3 Homogenised Matrix $[B^*]$ Derivatives

The expressions of $[B^*]$ derivatives are not provided since $[B^*] = [0]$.

B.2.4 Homogenised Matrix $[H^*]$ Derivatives

Derivative with respect to $R_{0K}^{A^*}$ Since there is no dependence between $R_{0K}^{A^*}$ and $[H^*]$ polar parameters, the derivative with respect to $R_{0K}^{A^*}$ is zero.

$$\left[\frac{\partial[A^*]}{R_{0K}^{A^*}} \right] = [0] \quad (\text{B.10})$$

Derivative with respect to $R_1^{A^*}$

$$\left[\frac{\partial[H^*]}{R_1^{A^*}} \right] = \begin{bmatrix} \frac{R}{R_1} \left[\cos 2(\Phi + \Phi_1 - \Phi_1^{A^*}) \right] & \frac{R}{R_1} \left[\sin 2(\cos 2(\Phi + \Phi_1 - \Phi_1^{A^*})) \right] \\ \frac{R}{R_1} \left[\sin 2(\cos 2(\Phi + \Phi_1 - \Phi_1^{A^*})) \right] & \frac{R}{R_1} \left[\cos 2(\cos 2(\Phi + \Phi_1 - \Phi_1^{A^*})) \right] \end{bmatrix} \quad (\text{B.11})$$

Derivative with respect to Φ_1^{A*}

$$\left[\frac{\partial[H^*]}{\Phi_{1A^*}} \right] = \begin{bmatrix} R_1^{A*} \frac{R}{R_1} 2sen \left[2(\Phi + \Phi_1 - \Phi_1^{A*}) \right] & -R_1^{A*} \frac{R}{R_1} 2cos \left[2(\Phi + \Phi_1 - \Phi_1^{A*}) \right] \\ -R_1^{A*} \frac{R}{R_1} 2cos \left[2(\Phi + \Phi_1 - \Phi_1^{A*}) \right] & -R_1^{A*} \frac{R}{R_1} 2sen \left[2(\Phi + \Phi_1 - \Phi_1^{A*}) \right] \end{bmatrix} \quad (\text{B.12})$$

Bibliography

- [1] Frediani A. Buttazzo G. *Variational analysis and aerospace engineering: mathematical challenges for the aerospace of the future. Springer optimization and its applications*. Springer, 2012.
- [2] Carrera E. *Lecture Notes of Nonlinear Aerospace Structures Analysis*. Polytechnic of Turin, 2010.
- [3] Schnell W. Eschenauers H. Olhoff N. *Applied Structural Mechanics, Fundamentals of Elasticity, Load-Bearing Structures, Structural Optimization*. Springer, 1997.
- [4] Verchery G. “Les invariants des tenseurs d’ordre 4 du type de l’élasticité”. In: *Proc. of colloque Euromech 115, Villard-de-Lans, France*. 1979.
- [5] Anderson C. J Kargahi M. “Structural Optimization for Seismic Design”. In: *13th World Conference on Earthquake Engineering*. 1979.
- [6] Montemurro M. “An extension of the polar method to the First-order Shear Deformation Theory of laminates”. In: *Composite Structures* 127 (2015), pp. 328–339.
- [7] Montemurro M. “Optimal Design of Advanced Engineering Modular Systems through a New Genetic Approach”. PhD thesis. UPMC, Paris VI, France, 2012.
- [8] Montemurro M. “The polar analysis of the Third-order Shear Deformation Theory of laminates”. In: *Composite Structures* 131 (2015), pp. 775–789.
- [9] Montemurro M. and Catapano A. “A new paradigm for the optimum design of variable angle tow laminates”. In: *Variational analysis and aerospace engineering: mathematical challenges for the aerospace of the future. Springer optimization and its applications*. Vol. 116. Springer International Publishing, 2016.
- [10] Montemurro M. and Catapano A. “On the effective integration of manufacturability constraints within the multi-scale methodology for designing variable angle-tow laminates”. In: *Composite Structures* 161 (2017), pp. 145–159.
- [11] Varadharajan M. M. “Optimization Of Structures: Shape and Weight Optimization of Structural Members”. In: *Research Gate* (2013).
- [12] Reddy N. J. *Mechanics of Laminated Composite Plates and Shells: Theory and Analysis, Second Edition*. CRC Press, 2003.
- [13] Vannucci P. *Anisotropic Elasticity*. Springer, 2017.

- [14] Greville T. “Theory and Applications of Spline Functions”. In: *Academic Press* (1969), pp. 1–36.
- [15] Piegl L. Tyler W. *The NURBS Book*. Springer, 1995.
- [16] Montemurro M. Vincenti A. Vannucci P. *BIANCA, A Genetic Algorithm for Engineering Optimisation - User guide*. HAL archives-ouvertes, 2011.
- [17] Nocedal J. Wright S.J. *Numerical Optimization*. Springer - Verlag, 2006.
- [18] Sun W. Yuan Y.-X. *Optimization Theory and Methods - Nonlinear Programming*. Springer, 2006, pp. 84–100.