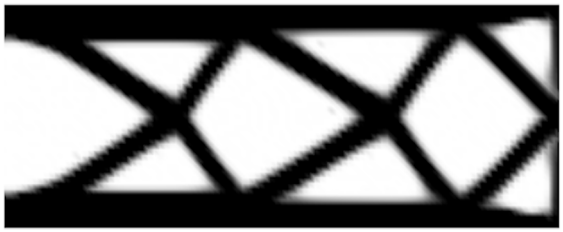
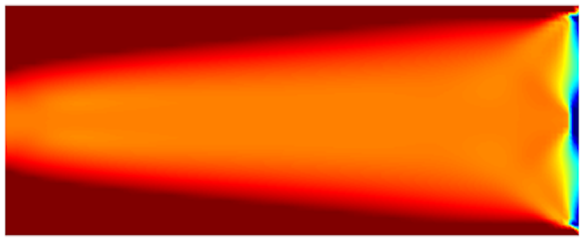
Topology optimization, variable stiffness design and staggered optimization for laminated composite structures

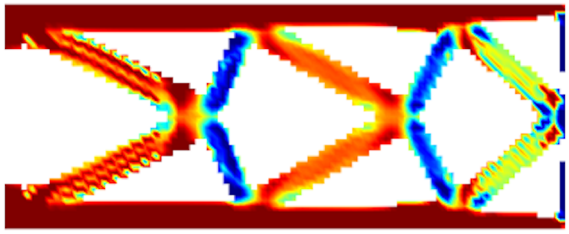
A COMPARATIVE STUDY

Master of Science Thesis

Nikoleta Pasvanti







TU Delft

Topology optimization, variable stiffness design and staggered optimization for laminated composite structures

A COMPARATIVE STUDY

by

Nikoleta Pasvanti

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Thursday August 26, 2021 at 09:30 AM.

Student number:	5141206	
Submitted on:	August 4, 2021	
Thesis committee:	Dr. S. R. Turteltaub,	TU Delft, supervisor
	ir. W. van den Brink,	NLR, co-supervisor
	Dr. ir. D. M. J. Peeters,	TU Delft
	Dr. C. D. Rans,	TU Delft

TU Delft

Abstract

The demand for efficient lightweight structures grows rapidly in the aerospace sector. Topology optimization, introduced in the late 80s, is a method capable of producing such structures and has been mainly studied for isotropic materials. On the other hand, the performance of composite structures, which are already widely employed in the aerospace industry, can be now improved due to the latest advances in manufacturing. Automated Fiber Placement has paved the way for further exploiting the capabilities of composites and led to the introduction of an optimization method, namely three-step variable stiffness design, that alters the fiber orientation within each ply, leading to a variable stiffness laminate. In this thesis, a staggered optimization method that simultaneously improves the material and the structural performance of balanced and symmetric laminated composite structures under planar loading is developed by coupling the two aforementioned methods.

A finite element code is developed based on the equations for a static and linearly elastic problem. For the topology optimization problem, the nodal density values are used as design parameters and the Solid Isotropic Material with Penalization approach is chosen, along with a density filtering. The first step of the variable stiffness design method is implemented, using the nodal lamination parameters as design variables. The coupling of the two optimization techniques is performed through a staggered optimization, where both techniques are implemented successively, at each iteration. A gradient-based scheme is used for both topology optimization and variable stiffness design and the steepest descent method is implemented for the design update. The sensitivity analysis is performed based on the continuous adjoint method.

Three different examples are demonstrated in this thesis; the case of a flat composite plate, a lug and an aircraft chair bracket. The whole optimization procedure, including pre- and post-processing, is carried out using an algorithm developed in MATLAB. The meshes are externally created and imported in the code. Convergence studies and a comparison with indicative examples from the literature are performed in order to verify the obtained results. A parametric analysis of the plate studies the effects of the different topology optimization parameters. Finally, comparative analyses are executed for the three aforementioned designs investigating both the structural and the computational efficiency of the results obtained with each of the three optimization methods, i.e. topology optimization, variable stiffness design and staggered optimization.

Acknowledgements

With this thesis, a big chapter of my life, being an MSc student at TU Delft, has come to an end. Studying abroad has been fascinating; I made a lot of friends and so many wonderful memories. Nevertheless, these two years have also been quite challenging. There have been lots of people by my side throughout this period and I would like to express my immense gratitude to them.

Firstly, I would like to thank my supervisor, Dr. Sergio Turteltaub, for his time, his continuous support and his investment to this project. I am very grateful for the valuable insights I gained from all the meetings that we had and his guidance to overcome the obstacles along the way. Secondly, I am thankful to my co-supervisor from NLR - Netherlands Aerospace Center, Wouter van den Brink, who gave me the opportunity to work on such an intriguing topic. Furthermore, I would like to express my gratitude to my parents, who supported and encouraged me throughout my whole academic career. Also, a big thank you goes to all the friends I made in Delft; all this progress the last two years would not have been possible without them. I owe a special thanks to Srinivasan Ganesh Ram, who has always been encouraging and believed in me, when I was not able to do that myself. Finally, I am grateful to my best friends in Greece, Chrysa and Dimitra, who, even from a distance, are always by my side.

Nikoleta Pasvanti
Delft, August 2021

Contents

List of Tables

List of Figures

Nomenclature

Abbreviations

AFP	Automated Fiber Placement
AM	Additive Manufacturing
BVP	Boundary Value Problem
CCSA	Conservative Convex Separable Approximations
CLPT	Classical Lamination Plate Theory
DOF	Degrees Of Freedom
FE	Finite Element
GUI	Graphical User Interface
SIMP	Solid Isotropic Material with Penalization

Symbols

$\mathbf{\Gamma_i}$	Material invariant matrices
$\boldsymbol{\varepsilon}^*$	Adjoint strain tensor
Δp	Density penalty power step size
Γ^e	Element boundary
λ, Λ	Lagrange multiplier
$\hat{\mathbf{C}}$	SIMP stiffness tensor
$\hat{\mathbf{t}}$	Traction vector
$\mathbf{A}$	In-plane stiffness matrix
$\mathbf{B}$	Coupling stiffness matrix
$\mathbf{D}$	Bending stiffness matrix
$\mathbf{F^e}$	Element force vector
$\mathbf{F}$	Global force vector
$\mathbf{J}$	Jacobian matrix
$\mathbf{K^e}$	Element stiffness matrix
$\mathbf{K}$	Global stiffness matrix
$\mathbf{n}$	Normal vector
$\mathbf{u}^*$	Adjoint displacement vector
$\mathbf{u}$	Displacement vector
$\mathbf{x}$	Position vector
$\mathscr{J}$	Jacobian
$\mathscr{L}$	Lagrangian function
ν_{ij}	Poisson's ratios $i-j$
Ω	Structural domain
$\partial\Omega, S$	Structural domain boundary
ψ_j	Shape function
ρ	Density
$\sigma, \boldsymbol{\sigma}$	Stress, stress tensor
θ	Fiber orientation angle
$\varepsilon, \boldsymbol{\varepsilon}$	Strain, strain tensor
ξ_I, η_J	Gauss points for surface integral
A	Area
a_i	Gauss points for line integral
E^e	Element Young's modulus
E_{ijkl}	Elasticity tensor components
E_i	Young's modulus in material direction i
G	Gradient
G_{12}	Shear modulus in the 1-2 plane
h	Laminate thickness
L	Objective functional augmented with the equilibrium constraint
$N, \mathbf{N}$	In-plane force resultant, in-plane force resultant vector
p	Density penalty power
Q_{ij}	Reduced stiffness matrix components
R	Resource constraint
$r_{\min}$	Minimum radius for the projection scheme
u_0	Midplane displacement towards X direction
U_i	Material invariant properties
V	Volume

v_0	Midplane displacement towards Y direction	w	Strain energy density
$V_{i\mathbf{A}}$	In-plane lamination parameters	w_i	Gauss weights for line integral
$V_{i\mathbf{B}}$	Coupling lamination parameters		
$V_{i\mathbf{D}}$	Bending lamination parameters	W_I, W_J	Gauss weights for surface integral

1

Introduction

Lightweight structures, providing high specific stiffness and strength, are widely employed in the aerospace sector due to their indisputable efficiency and high savings. One method that produces such lightweight structures is topology optimization. Its objective is to optimize the structural performance of a part usually made of an isotropic material, achieving the minimum weight possible. Since the introduction of the method as we know it today, in the late 80s, it has evolved using different approaches to achieve its target. Solid Isotropic Material with Penalization (SIMP) (Bendsøe, 1989; Bendsøe & Sigmund, 1999; Hassani et al., 2012; Sigmund, 1997), phase field (Wallin et al., 2012), level set (Allaire et al., 2004; Allaire et al., 2002; M. Wang et al., 2003) and evolutionary (Xie & Steven, 1993; Young et al., 1999) are some of the most well-known techniques to deal with the topology optimization problem. However, the SIMP technique seems to have obtained a leading position among the rest, after its wide acceptance by the academic community and its employment in most of the commercial finite element codes. Its formulation is simple and it manages to produce useful results, in a computationally efficient time framework (Rozvany, 2001, 2009; Sigmund & Maute, 2013). Due to significant research effort in the last three decades, the topology optimization problem has overcome the numerical issues it was facing and has kept gaining popularity among engineering practices.

Another relatively new and promising optimization problem is the one that seeks the optimal material distribution in fiber reinforced composite structures. These structures constitute one of the best options for lightweight design since they provide the designer with the freedom to tailor their properties. Commercial aircraft such as the Airbus A350 XWB and the Boeing 787 incorporate composite materials for more than half of their structural weight (Albazzan et al., 2019). The recent developments in manufacturing technologies, such as the Automated Fiber Placement (AFP) and the Additive Manufacturing (AM) of short and continuous fibers, paved the way for the exploitation of optimization techniques in composite structures. Due to the freedom of placing fibers in any direction, the research concentrated on developing methods to optimize the performance of the structures by varying the fiber orientations within each ply. Towards this direction, the three-step variable stiffness design was proposed (IJsselmuiden, 2011). These three steps, which constitute three different procedures to obtain the variable stiffness laminate, are the laminate stiffness optimization (Setoodeh, Abdalla, et al., 2006a), the stacking sequence retrieval (Peeters, Hesse, et al., 2015; Setoodeh, Blom, et al., 2006; van Campen et al., 2012) and the fiber path construction (Blom et al., 2010; van Tooren & Elham, 2016; Wu et al., 2015).

Nowadays, in order to achieve the production of lightweight and, at the same time, even more enhanced, in terms of performance, structures, efforts have been made towards designing topology optimized variable stiffness parts. However, the majority of the existing literature focuses on optimizing the topology and the fiber orientation of one ply rather than a laminate of an unknown stacking sequence. Thus, the research objective that will contribute to future developments towards that direction can be formed as:

The research objective of this thesis is to develop an *optimization method* that *simultaneously improves* the *material* and *structural performance* of *laminated composite structures* used for aerospace applications. This objective is achieved by the *coupling* of two optimization techniques, namely *topology optimization* and

variable stiffness design.

This report is structured as follows; the literature review is presented in chapter §2. The methodology used in this thesis is described in chapter §3 and the numerical implementation of the methodology follows in chapter §4. The results are presented and discussed in chapter §5 and, finally, conclusions and future work recommendations are addressed in chapter §6.

2

Literature Review

This chapter provides an overview on the research topics of topology optimization and variable stiffness design. The theoretical background on structural optimization is briefly presented in section §2.1. The evolution of topology optimization is discussed in section §2.2 and the variable stiffness design method is covered in section §2.3. The coupling of the two optimization approaches for laminates of an unknown stacking sequence is addressed in section §2.4 and a synopsis is provided in section §2.5.

2.1. Fundamentals of structural optimization

2.1.1. The optimization problem

An optimization problem is consisted of an objective function f that should be either minimized or maximized. This way the notion of optimality is achieved. The problem is minimized/maximized with respect to certain parameters, namely design variables, denoted as $\mathbf{y}$. The branch of mathematics that deals with seeking the optimal design variables is called mathematical programming. When the objective function is subject to certain requirements, known as constraints, the optimization is called a constrained optimization. These constraints can be formed as equalities and/or inequalities. Thus, the formulation of the optimization problem (Haftka & Gürdal., 1993) is

$$\min \text{ or } \max f(\mathbf{y}), \tag{2.1.1}$$

s.t.

$$g_j(\mathbf{y}) \geq 0 \quad j = 1, \ldots, n_g, \tag{2.1.2}$$

$$h_k(\mathbf{y}) = 0 \quad k = 1, \ldots, n_e, \tag{2.1.3}$$

where n_g is the number of inequality constraints, characterized by the scalar functions g_j, and n_e is the number of equality constraints, characterized by the scalar functions h_k.

The design variables used can be either continuous or discrete and this choice depends mainly on the optimization problem to be solved.

An optimization problem can be either linear or nonlinear. The former means that the objective function and the constraints, also called "functions of interest" (Martins, 2012), are expressed as linear functions of the design parameters. The latter means that one or more functions of interest vary nonlinearly with respect to the design variables (Haftka & Gürdal., 1993).

The design space is split in two domains, the feasible and the infeasible domain. The design variables that belong in the feasible domain satisfy all the constraints of the problem. Even if one constraint is violated, the design variable falls under the infeasible domain category. An inequality constraint is called active when it is equal to zero and inactive when the inequality is satisfied.

2.1.2. Constrained optimization

In an optimization problem it is crucial to ensure that the optimum found is a global optimum. Constrained optimization problems make use of the Lagrange multipliers method to find a solution. For the case of equality constraints only, the objective function f incorporates the constraints $h_j(\mathbf{y}) = 0$ multiplied by the Lagrange multipliers λ_j (Haftka & Gürdal., 1993), with $j = 1, \ldots, n_e$, forming the Lagrangian function

$$\mathscr{L}(\mathbf{y}, \boldsymbol{\lambda}) = f(\mathbf{y}) + \sum_{j=1}^{n_e} \lambda_j h_j. \tag{2.1.4}$$

The conditions

$$\frac{\partial \mathscr{L}}{\partial y_i} = \frac{\partial f}{\partial y_i} - \sum_{j=1}^{n_e} \lambda_j \frac{\partial h_j}{\partial y_i} = 0 \quad i = 1, \ldots, n, \tag{2.1.5}$$

$$\frac{\partial \mathscr{L}}{\partial \lambda_j} = h_i(\mathbf{y}) = 0 \quad j = 1, \ldots, n_e, \tag{2.1.6}$$

where n is the number of design variables, are the necessary conditions for a stationary regular point, in case the gradients of the constraints $\nabla h_j(\mathbf{y})$ are linearly independent.

In case of inequality constraints, the Lagrangian is written with the assistance of slack variables t_j (Haftka & Gürdal., 1993) as

$$\mathscr{L}(\mathbf{y}, \mathbf{t}, \boldsymbol{\lambda}) = f - \sum_{j=1}^{n_g} \lambda_j \left(g_j - t_j^2 \right). \tag{2.1.7}$$

The necessary conditions for a stationary regular point (Haftka & Gürdal., 1993) are given by the gradients of the Lagrangian with respect to the design variables y_i, the Lagrange multipliers λ_j and the slack variables t_j

$$\frac{\partial \mathscr{L}}{\partial y_i} = \frac{\partial f}{\partial y_i} - \sum_{j=1}^{n_g} \lambda_j \frac{\partial g_j}{\partial y_i} = 0 \quad i = 1, \ldots, n, \tag{2.1.8}$$

$$\frac{\partial \mathscr{L}}{\partial \lambda_j} = -g_j + t_j^2 = 0 \quad j = 1, \ldots, n_g, \tag{2.1.9}$$

$$\frac{\partial \mathscr{L}}{\partial t_j} = 2\lambda_j t_j = 0 \quad j = 1, \ldots, n_g. \tag{2.1.10}$$

According to the Kuhn-Tucker conditions, which constitute the necessary conditions for a minimum, (2.1.8) should be satisfied and in case of an inactive constraint the corresponding Lagrange multiplier should be equal to zero.

The Kuhn-Tucker conditions are also sufficient to prove that an optimum is a global optimum when the number of active constraints is the same as the number of design variables or when the optimization problem is convex.

In general, an optimization problem is convex when the objective function $f_0 : \mathbf{R}^n \to \mathbf{R}$ and the constraints $f_i : \mathbf{R}^n \to \mathbf{R}$ $(i = 1, \ldots, m)$ are convex (Boyd & Vandenberghe, 2004), meaning that they satisfy

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y), \tag{2.1.11}$$

for all $x, y \in \mathbf{R}^n$ and all $\alpha, \beta \in \mathbf{R}$ with $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$.

Linear programming optimization problems are convex. However, most optimization problems are non-convex and, for this reason, convex approximations are usually used to obtain the solution.

2.1.3. Design update

An optimization problem requires a certain number of iterations until convergence to the optimum solution. In each iteration the design is updated. The search that is performed for the design update depends on the method that is used. Two popular methods in the literature are the so-called gradient-based and direct search methods (Arora, 2012).

For the purpose of this thesis, a gradient-based search will be used, where the gradients of the problem functions determine the direction of the design update (Arora, 2012). There are two essential calculations that accompany a gradient-based method; firstly, the search direction has to be computed and secondly, the step size towards that search direction has to be estimated. A couple of popular search direction techniques are the steepest descent and the conjugate gradient method.

To obtain the problem gradients for the gradient-based optimization, it is necessary to compute the design sensitivities, i.e. the derivatives of the governing equations with respect to the design variables. These derivatives are obtained through a procedure called sensitivity analysis.

In this thesis, a continuous adjoint method is used for the sensitivity analysis. The term continuous means that the analytical method is performed in the continuous setting and after the derivatives with respect to the design parameters are obtained, the discretization for the finite element model is performed. The term adjoint is referred to the way the derivative of the state variables with respect to the design parameters, also known as implicit sensitivity, is obtained. This approach is preferred in this case, due to the fact that only one FE (Finite Element) problem needs to be solved in each optimization iteration, rendering the method more efficient than its alternatives. The sensitivity analysis performed using the continuous adjoint method is described in detail in chapter §3.

2.2. Topology optimization

2.2.1. Definition

Topology optimization is an engineering optimization problem that aims to improve the structural performance of a specified design domain Ω by distributing the material within the domain in an optimal manner. This is achieved by minimizing an objective function f, which is equal to the strain energy of the structure in case of compliance minimization, with respect to the density function $\rho(\mathbf{x})$. The density function is the design variable at a position $\mathbf{x}$ inside the domain Ω. It can either take the value 0 to indicate absence of material or 1 to indicate presence of material within the domain. The problem is subject to a volume constraint $\int_\Omega \rho(\mathbf{x}) dV \leq R$, where R is the desirable volume, known as the resource constraint, and possibly additional constraints (Sigmund & Maute, 2013).

The topology optimization problem in its nature uses discrete design variables and suffers from lack of solutions. Using continuous density design variables renders the convergence of the optimization problem possible. The approach that is widely used in the literature using continuous design variables is the density approach, also known as Solid Isotropic Material with Penalization (SIMP) (Bendsøe, 1989; Mlejnek, 1992; Zhou & Rozvany, 1991). The topology optimization problem is discretized using a fixed mesh and nodal or element density values are used as design variables (Sigmund & Maute, 2013).

2.2.2. The SIMP method

Right before the so-called SIMP method was introduced, Bendsøe and Kikuchi (1988) published a paper focused on using homogenization approaches for structural topology optimization. Their goal was to present a method that computes the optimal material distribution of a structure. They suggested the use of composite materials, making the utilization of a continuous density function in the interval $[0, 1]$ feasible. These composite materials were periodic cells that contributed to the representation of the intermediate densities. The notion of homogenization was applied for the computation of the effective properties of the cells and the minimum compliance problem was solved using the density and the rotation angle of the cell as design variables.

A subsequent publication by Bendsøe (1989) focused on converting the discrete nature of the topology optimization problem as we know it today to a continuous one, following the steps of the previous paper by Bendsøe and Kikuchi (1988). That was when the power law was introduced for the penalization of the intermediate densities, called the "direct approach" at the time. Bendsøe (1989) expressed the elasticity tensor E_{ijkl} in the design domain Ω as a function of the penalized density $[\rho(\mathbf{x})]^p$, where p is the penalty power, multiplied by the constant elasticity tensor of the material $\bar{E}_{ijkl}$

$$E_{ijkl}(\mathbf{x}) = [\rho(\mathbf{x})]^p \bar{E}_{ijkl}, \tag{2.2.1}$$

with $\mathbf{x} \in \Omega,\ 0 \leq \rho(\mathbf{x}) \leq 1,\ p \gg 1$.

This approach was considered as an artificial material approach since the intermediate densities get penalized and, this way, solutions become closer to the values 0 and 1 (Figure 2.1). These solutions are commonly known in the literature as "black and white". The author was not in favor of the power law at the time, due to the absence of physical interpretation of the material used and the mesh dependency of the approach.

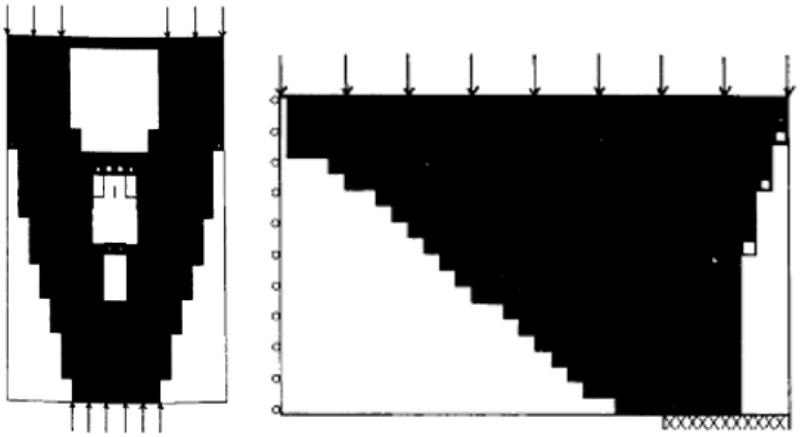


Figure 2.1: Numerical examples generated by the "direct approach" with p = 4 (Bendsøe, 1989).

The power law was used again by Sigmund (1997), when he introduced an approach to obtain the optimal design of compliant mechanism topologies. He allowed the use of the material density in each element ρ^e as a continuous design variable in the interval $[\rho_{\min}, 1]$. The use of a lower limit $\rho_{\min}$ is to avoid the stiffness matrix being singular in the FE problem. The power law was used for the Young's modulus of each element E^e in the form

$$E^e = \left(\rho^e\right)^p E^0 \quad e = 1, \ldots, N, \tag{2.2.2}$$

with E^0 being the material's Young's modulus and $p = 3$ to sufficiently penalize intermediate densities.

Two years later, Bendsøe and Sigmund (1999) presented a physical interpretation of the black and white solutions derived from the SIMP method by introducing interpolation functions for the material properties. They presented that, for isotropic materials, the SIMP model can comply with the Hashin-Shtrikman bounds, thus any SIMP stiffness can be interpreted by a composite made up with void and material. The geometry of these composites, else microstructures (Figure 2.2), can be obtained using inverse homogenization.

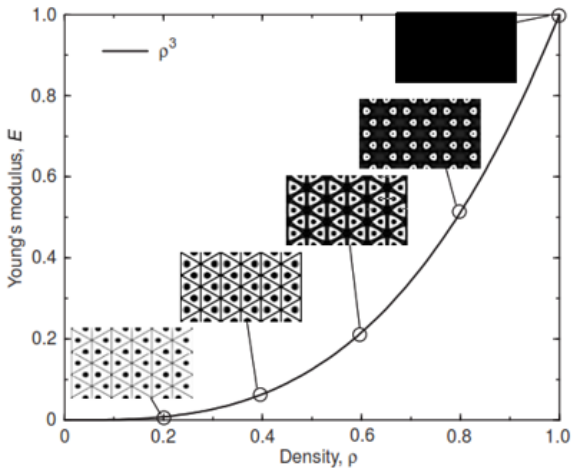


Figure 2.2: Interpretation of the SIMP material properties for $p = 3$ and $\nu = 1/3$ using microstructures (Bendsøe & Sigmund, 1999).

.

Later, Sigmund (2001) published a 99-line MATLAB code for topology optimization of statically loaded structures. The SIMP approach was implemented for the minimum compliance problem and sensitivity filtering (Sigmund, 1997) was used to ensure mesh independency of the solution. The result for the popular cantilever beam example using this code is presented in Figure 2.3.

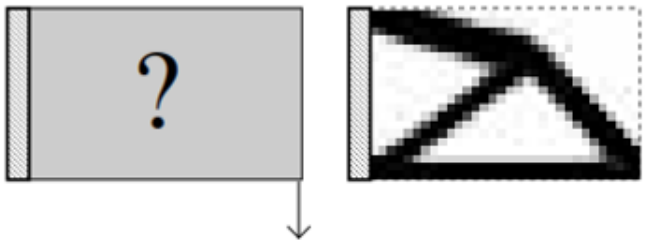


Figure 2.3: Cantilever beam example. Design domain and boundary conditions (left) and solution (right) (Sigmund, 2001).

An extension of the 99-line code (Sigmund, 2001) was developed by Andreassen et al. (2011). This code is an 88-line one, more computationally efficient than its predecessor and includes the option of density filtering (Bourdin, 2001; Bruns & Tortorelli, 2001), next to the sensitivity filtering. A modified SIMP approach was used in this case, using a minimum value for the material's Young's modulus $E_{\min}$, as

$$E^e\left(\rho^e\right) = E_{\min} + \left(\rho^e\right)^p \left(E_0 - E_{\min}\right) \quad \rho^e \in [0,1], \tag{2.2.3}$$

which was considered advantageous over the classical SIMP approach (Sigmund, 2007).

Both filters used, either on sensitivities or densities, suppress checkerboards (Díaz & Sigmund, 1995) and prevent mesh dependency. A number of alternative implementations are also presented in Andreassen et al. (2011), such as the convolution-based approach for density filtering and the Helmholtz type partial differential equation for sensitivity/density filtering, which have less memory requirements.

The isogeometric analysis has also been used for topology optimization, incorporating the SIMP model (Hassani et al., 2012). Using this type of analysis, any field variable component corresponds to a surface. This surface can be constructed using Non-uniform Rational Basis Splines (NURBS) and the control points of the splines are used for the discretization of the domain. The geometry, the displacements and the density function are approximated using the basis functions of NURBS. The numerical examples proved that increasing the number of control points, a decrease in gray areas is achieved as well as improved compliance values.

Also, it was shown that using isogeometric analysis instead of FE modelling provides results that are independent on the discretization, without any checkerboarding issues.

There are multiple other approaches used for topology optimization in the literature; topological derivatives (Novotny et al., 2003; Sokolowski & Zochowski, 1999), the phase field approach (Wallin et al., 2012), the level set approach (Allaire et al., 2004; Allaire et al., 2002; M. Wang et al., 2003) with multiple variations, as well as methods that solve the optimization problem using discrete design variables, such as Evolutionary Structural Optimization (ESO) (Xie & Steven, 1993) and Bidirectional Evolutionary Structural Optimization (BESO) (Young et al., 1999). So far, the most prominent one, other than the density approach, is the level set method. A few level set methods manage to reproduce the inverter example, introduced in Sigmund (1997), shown in Figure 2.4, correctly. This solution is quite challenging to reproduce and has been chosen as a benchmark problem for topology optimization (Sigmund, 2007; Sigmund, 2009). However, despite its broad application, it is still unclear if and in which cases it can be preferred over the density approach (Sigmund & Maute, 2013).

Figure 2.4: Inverter example (Sigmund, 2009).

Using the SIMP approach with a penalization power $p = 1$ the compliance minimization problem is convex and offers a unique solution (Bendsøe & Sigmund, 2004; Sigmund & Maute, 2013). Penalizing the intermediate densities with $p > 1$ converts the convex problem to non-convex. This means that the solution might be a local optimum. However, using a continuation method (Sigmund & Petersson, 1998) by gradually increasing p from 1 to a value $p \geq 3$ the solution found cannot be driven far from the global optimum (Rozvany, 2001, 2009).

The implementation of SIMP is straightforward and the method can be applied to a wide range of problems. Its results are approved by the academic community, it is computationally efficient and, therefore, most of the commercial software have implemented it (Rozvany, 2001). The comparison of the different methods depends to a certain extent on the preferences of the user. Rozvany (2009) emphasizes the different needs between the academia and the industry; the former prefers the theoretically optimal solution with little parameter tuning whereas the latter favors little computational effort, general applicability of the method and simpler topologies.

2.2.3. Numerical issues and workaround techniques

The use of filtering techniques is crucial for density problems (Sigmund & Maute, 2013) in order to tackle mesh dependency and checkerboarding (F. Wang et al., 2011).

Sigmund (1997) suggested a filter imposed on the element sensitivities, using a factor that depends on the neighboring elements within a filter radius r_{min} from the element. Later, Sigmund and Petersson (1998) published a review on the numerical instabilities in topology optimization and the available solutions at the time. One of the most popular issues mentioned is checkerboarding, that is the formation of alternating black and white patterns as shown in Figure 2.5. The second well-known problem is mesh dependency, meaning that the topology of the structure changes with mesh refinement instead of being held constant and becoming better refined. The authors indicate that the filtering techniques are able to tackle the above, but constitute a heuristic approach.

Figure 2.5: The issue of checkerboarding (Sigmund & Petersson, 1998).

Acquiring different solutions to the same problem is known as the local minima pathology and it can be detected by altering the algorithm parameters such as the initial design. Local minima can be prevented using continuation methods, as previously mentioned.

The density filtering was introduced later by Bourdin (2001) and Bruns and Tortorelli (2001). Here, the element densities are modified, instead of the element sensitivities. Their value is modified using a weight function, which again depends on the neighboring elements within a radius $r_{\min}$ (Guest et al., 2004; Sigmund, 2007; Sigmund, 2009; F. Wang et al., 2011).

The above filtering techniques guaranteed global convergence but not the elimination of gray transitional areas in the solution (F. Wang et al., 2011). A couple of notable efforts to obtain total black and white solutions are the use of a Heaviside step function for the element densities (Guest et al., 2004) and morphology-based filters (Sigmund, 2007).

2.3. Variable stiffness design

Composite structures, traditionally, are consisted of a specific number of plies which are given a constant fiber orientation angle. Due to the recent evolution of new technologies such as AM and AFP, there have been emerging efforts to optimize their performance according to the specified application. These efforts mainly study the variable stiffness laminates, where the fiber orientation angle varies within each ply. These laminates can be modelled in three ways; by assigning discrete fiber angles at every coordinate, by a fiber path constructed using a curvilinear function or by stiffness using lamination parameters.

Albazzan et al. (2019) listed the pros and cons for each modeling technique. Using discrete fiber angles, the entire design space is used but the problem is non-linear and non-convex, computationally inefficient and a step of post-processing is needed to obtain the fiber path for manufacturing. The fiber path technique constructs smooth and continuous paths and facilitates the imposition of the curvature constraint for manufacturing. However, it is also non-linear, non-convex, the function used limits the design space and a different function is needed for different developable surface. Finally, modelling with lamination parameters makes use of the whole design space, the feasible region of the design variables is convex and the number of layers does not depend on the number of the design variables. On the other hand, a significant post-processing is required to retrieve useful information such as the fiber angles and in case combined stiffness properties have to be used, both in-plane and flexural, a feasible region needs to be defined for both.

A multi-step optimization approach (IJsselmuiden, 2011) has been deemed efficient to deal with the variable stiffness design of composite structures. This approach comprises three steps. The first one is the laminate stiffness optimization, where the stiffness is optimized in terms of lamination parameters. The second one is the stacking sequence retrieval, where the fiber orientation angles are retrieved from the stiffness properties. Finally, in the last step, the fiber path construction, the continuous paths are created for manufacturing using the information of the previous step. These three steps are illustrated in Figure 2.6. This thesis focuses on step 1, which is described in detail in subsection §2.3.1. Steps 2 and 3 are summarized in subsections §2.3.2 and §2.3.3.

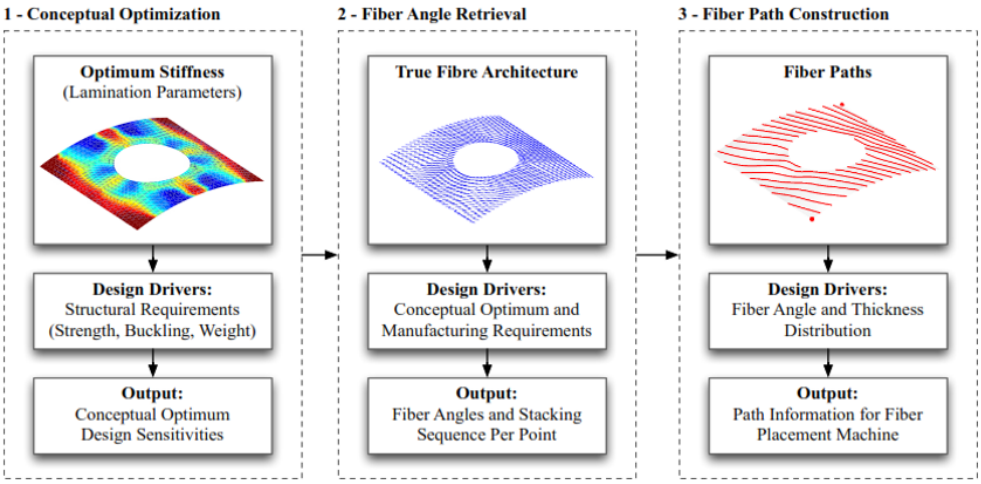


Figure 2.6: The three-step optimization approach for variable stiffness laminates (IJsselmuiden, 2011).

2.3.1. Step 1 - laminate stiffness optimization

Stiffness representation by lamination parameters

There are twelve lamination parameters overall that represent the stiffness of a laminate (Tsai & Hahn, 1980). These are the in-plane $V_{i\mathbf{A}}$, coupling $V_{i\mathbf{B}}$ and bending parameters $V_{i\mathbf{D}}$

$$\begin{aligned}
\left(V_{1\mathbf{A}}, V_{2\mathbf{A}}, V_{3\mathbf{A}}, V_{4\mathbf{A}}\right) &= \int_{-\frac{1}{2}}^{\frac{1}{2}} (\cos 2\theta, \sin 2\theta, \cos 4\theta, \sin 4\theta)\, d\bar{z}, \\
\left(V_{1\mathbf{B}}, V_{2\mathbf{B}}, V_{3\mathbf{B}}, V_{4\mathbf{B}}\right) &= 4\int_{-\frac{1}{2}}^{\frac{1}{2}} \bar{z}(\cos 2\theta, \sin 2\theta, \cos 4\theta, \sin 4\theta)\, d\bar{z}, \\
\left(V_{1\mathbf{D}}, V_{2\mathbf{D}} V_{3\mathbf{D}}, V_{4\mathbf{D}}\right) &= 12\int_{-\frac{1}{2}}^{\frac{1}{2}} \bar{z}^2(\cos 2\theta, \sin 2\theta, \cos 4\theta, \sin 4\theta)\, d\bar{z},
\end{aligned} \tag{2.3.1}$$

where $\bar{z} = z/h$ is the normalized through the laminate thickness h coordinate and θ is the fiber orientation at $\bar{z}$.

The in-plane **A**, coupling **B** and bending **D** matrices of the **ABD** matrix vary linearly with respect to the lamination parameters

$$\begin{aligned}
\mathbf{A} &= h\left(\mathbf{\Gamma_0} + \mathbf{\Gamma_1} V_{1\mathbf{A}} + \mathbf{\Gamma_2} V_{2\mathbf{A}} + \mathbf{\Gamma_3} V_{3\mathbf{A}} + \mathbf{\Gamma_4} V_{4\mathbf{A}}\right), \\
\mathbf{B} &= \frac{h^2}{4}\left(\mathbf{\Gamma_1} V_{1\mathbf{B}} + \mathbf{\Gamma_2} V_{2\mathbf{B}} + \mathbf{\Gamma_3} V_{3\mathbf{B}} + \mathbf{\Gamma_4} V_{4\mathbf{B}}\right), \\
\mathbf{D} &= \frac{h^3}{12}\left(\mathbf{\Gamma_0} + \mathbf{\Gamma_1} V_{1\mathbf{D}} + \mathbf{\Gamma_2} V_{2\mathbf{D}} + \mathbf{\Gamma_3} V_{3\mathbf{D}} + \mathbf{\Gamma_4} V_{4\mathbf{D}}\right),
\end{aligned} \tag{2.3.2}$$

with material invariant matrices $\mathbf{\Gamma_i}$ defined in Abdalla et al. (2007).

For symmetric laminates, we have

$$V_{i\mathbf{B}} = 0. \tag{2.3.3}$$

Also, assuming balanced laminates, with limited bending-twisting coupling, we have

$$V_{2\mathbf{A}} = V_{4\mathbf{A}} = 0 \tag{2.3.4}$$

and

$$V_{2\mathbf{D}} = V_{4\mathbf{D}} \approx 0, \tag{2.3.5}$$

respectively.

In order for the lamination parameters to produce meaningful laminate stiffness matrices, a feasible domain or else, feasible range, is defined. This domain, for the in-plane lamination parameters, is convex and defined as (Hammer et al., 1997)

$$\begin{gathered} 2V_{1\mathbf{A}}^2\left(1-V_{3\mathbf{A}}\right)+2V_{2\mathbf{A}}^2\left(1+V_{2\mathbf{A}}\right)+V_{3\mathbf{A}}^2+V_{4\mathbf{A}}^2-4V_{1\mathbf{A}}V_{2\mathbf{A}}V_{4\mathbf{A}}\leq 1, \\ V_{1\mathbf{A}}^2+V_{2\mathbf{A}}^2\leq 1, \\ -1\leq V_{i\mathbf{A}}\leq 1 \quad (i=1,\ldots,4), \end{gathered} \tag{2.3.6}$$

which, for balanced and symmetric laminates, simplifies to

$$\begin{gathered} V_{3\mathbf{A}}\geq 2V_{1\mathbf{A}}^2-1, \\ -1\leq V_{i\mathbf{A}}\leq 1 \quad (i=1,3). \end{gathered} \tag{2.3.7}$$

One problem regarding the use of lamination parameters for optimization is the fact that there is no analytical expression of the feasible region combining all twelve parameters, because most of the available expressions include either only in-plane or only flexural parameters (Albazzan et al., 2019). Setoodeh, Abdalla, et al. (2006b) approximated the boundary of the general domain with numerous inequalities to be incorporated in the problem formulation, without restricting the fiber angles. The region for a predefined group of angles was later derived by Bloomfield et al. (2009).Wu et al. (2013, 2015) derived an explicit region for the four lamination parameters of orthotropic, balanced symmetric laminates. These nonlinear equations are considered the most efficient ones for the optimization of orthotropic laminates (Albazzan et al., 2019). There has also been an extension of Wu et al.'s work which incorporated the extension-shear and bending-twisting coupling (Raju et al., 2014).

Considering the planar case, the lamination parameters can be physically interpreted by looking at the $\mathbf{\Gamma_i}$ matrices and the elements of the in-plane stiffness matrix that they affect. For $V_{i\mathbf{A}}=0$, we get the in-plane stiffness matrix $\mathbf{A}$ of a quasi-isotropic laminate. Taking this into account, for a balanced laminate $\left(V_{2\mathbf{A}}=V_{4\mathbf{A}}=0\right)$, when $V_{1\mathbf{A}}$ tends to 1, the laminate comprises more fibers aligned with the 0° direction. On the other hand, when $V_{1\mathbf{A}}$ tends to -1, 90° fibers are prominent. In a similar manner, when $V_{3\mathbf{A}}$ tends to 1, there is a larger ratio of 0° and 90° fibers in the laminate and when it tends to -1, most of the fibers align with the 45° orientation (IJsselmuiden, 2011). The feasible region of in-plane lamination parameters for balanced and symmetric laminates, given in (2.3.7), is illustrated in Miki's diagram in Figure 2.7.

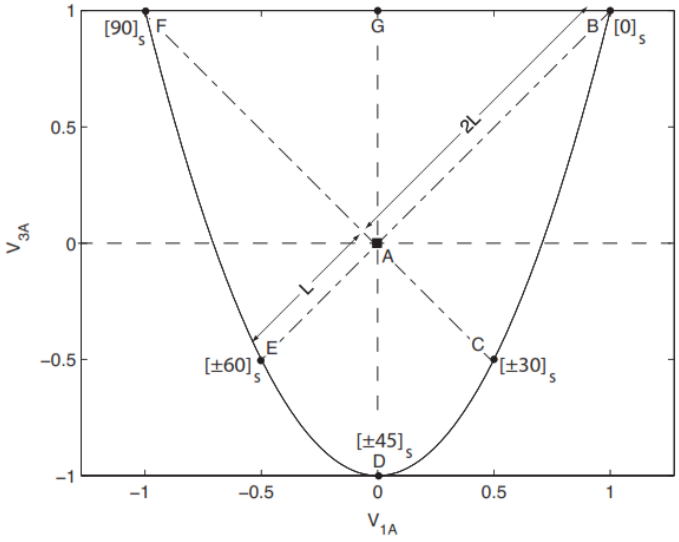


Figure 2.7: Miki's diagram for the in-plane lamination parameter feasible domain of balanced and symmetric laminates (Miki, 1982).

The first step in variable stiffness design

In the first step of the three-step variable stiffness design approach, the theoretical optimum, which depends on the application, is obtained using lamination parameters. Two optimization approaches were developed for this step using gradient-based methods; the FE analysis and the isogeometric analysis-based approach (Albazzan et al., 2019). This thesis is focused on the first one.

Hammer et al. (1997) sought the optimal layup of laminated composites for minimum compliance, and, instead of working directly with the ply thickness, the fiber orientations and the stacking sequence, they chose to use the lamination parameters. It is emphasized that the combination of ply thicknesses and angles that correspond to a lamination parameter distribution is not unique. Using the lamination parameters as design variables, the minimum compliance problem can be formulated. This is a convex problem and the result does not depend on the initial design, avoiding thus local optima.

Subsequently, Setoodeh, Abdalla, et al. (2006a) studied the optimal design of laminates for minimum compliance under in-plane or bending loads for symmetric laminates. Bilinear Classical Lamination Plate Theory elements (CLPT) were used and five different laminate cases were examined.

The compliance minimization problem for variable stiffness laminates at any domain node i $\left(i = 1, \ldots, n_n\right)$ is formulated as

$$\min_{\mathbf{v}_i} \frac{1}{2} \mathbf{N}_i^T \cdot \mathbf{A}^{-1}\left(\mathbf{V}_i\right) \cdot \mathbf{N}_i \quad \text{s.t. } (C1), \tag{2.3.8}$$

where $\mathbf{N}$ is the resultant forces vector, $\mathbf{V}$ is the nodal lamination parameter vector and $(C1)$ is the feasible region as defined in (2.3.6).

As already mentioned, this is a convex problem, as both the objective function and the constraints are convex.

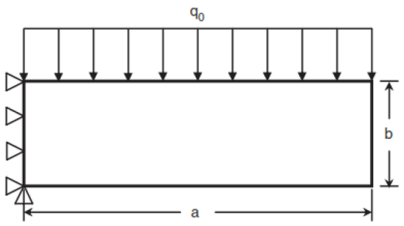


Figure 2.8: Cantilever plate with uniform load (Setoodeh, Abdalla, et al., 2006a).

The in-plane example used was a cantilever plate with a uniform loading as shown in Figure 2.8. The single layer variable stiffness result, that depicts the distribution of the fiber orientations obtained optimizing the angles directly, is shown in Figure 2.9. The prominent discontinuities are present due to the fact that this optimization, which uses fiber angles as design variables, is non-convex. The continuous lamination parameter distribution for the general variable stiffness laminate case, obtained optimizing the lamination parameters, is shown in Figure 2.10. The compliance is reduced by 36% with respect to the optimal constant stiffness laminate and 9% with respect to the variable stiffness lamina.

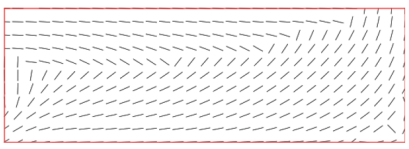


Figure 2.9: Optimal fiber angle distribution for the single layer variable stiffness case (Setoodeh, Abdalla, et al., 2006a).

In this first optimization step, continuity and smoothness for manufacturability are guaranteed by associating the lamination parameters with nodes in the FE analysis and automatically by the nature of the problem in isogeometric analysis (Albazzan et al., 2019). The minimum turning radius is an important constraint that

needs to be included in the optimization, because severe curvature can cause wrinkling of the inner tow during fiber steering. The constraint has been implemented in the lamination parameter space indirectly, by imposing an upper bound on the lamination parameters (Hong et al., 2020). However, the appropriate choice of this bound cannot be done in advance. Robustness of the laminate can be ensured by uncoupling of the in-plane and bending responses, simplifying a number of procedures, including analysis and manufacturing (Albazzan et al., 2019). The 10% rule has also been implemented by a couple of research teams (Abdalla et al., 2009; Albazzan et al., 2018).

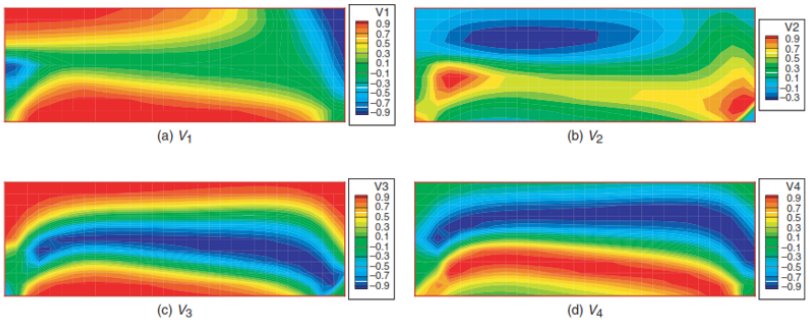


Figure 2.10: Optimal lamination parameter distributions for the general variable stiffness case (Setoodeh, Abdalla, et al., 2006a).

2.3.2. Step 2 - stacking sequence retrieval

The retrieval of the fiber angles is no longer a convex problem and various methodologies like curve fitting and evolutionary algorithms are used to deal with it (Albazzan et al., 2019). A particular emphasis on the implementation of the curvature constraint is given in this step.

In a subsequent work by Setoodeh, Blom, et al. (2006), a curvilinear path that represents the optimal set of in-plane lamination parameters is sought. The fiber orientation is expressed using basis functions of which the unknown coefficients $a_{ij}^{(k)}$ should be computed for each layer k. For the problem formulation, a rectangular domain of a balanced and symmetric panel is assumed with an already obtained lamination parameter distribution. The fiber orientation angle $\tilde{\theta}$ for an assumed layup $\left[\pm\theta_1 \pm \theta_2 \ldots \pm \theta_{n_q}\right]_s$, with number of layers $n_h = 2n_q$, in each layer k can be approximated as

$$\begin{aligned} \tilde{\theta}^{(k)}(\xi,\eta) &= \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} a_{ij}^{(k)} \hat{L}_i(\xi)\hat{L}_j(\eta), \\ k &= 1,2,\ldots,n_q, \end{aligned} \tag{2.3.9}$$

where m and n are the number of the basis functions in the normalized coordinates for the rectangular domain (ξ,η), $a_{ij}^{(k)}$ the design variables and $\hat{L}_i(\xi) = L_i(\xi)/\bar{L}_i$ is the normalized Lobatto polynomial.

Assuming balanced and symmetric laminates, the lamination parameters can be calculated from the fiber angle approximation $\tilde{\theta}^{(k)}$ at a layer k as

$$\begin{aligned} \tilde{V}_1(\xi,\eta) &= \frac{1}{n_q}\sum_{k=1}^{n_q} \cos\left(2\tilde{\theta}^{(k)}(\xi,\eta)\right), \\ \tilde{V}_3(\xi,\eta) &= \frac{1}{n_q}\sum_{k=1}^{n_q s} \cos\left(4\tilde{\theta}^{(k)}(\xi,\eta)\right). \end{aligned} \tag{2.3.10}$$

The problem, then, is formulated as a nonlinear least-square problem. The target is to find the fiber orientations that correspond to the closest lamination parameters to the optimal lamination parameter distribution,

whereas, at the same time, the maximum curvature constraint $\kappa_{\max}$ should be satisfied

$$\begin{gathered}\min_{\mathbf{X}} \frac{1}{2}\sum_{i=1}^{2n_n} f_i^2(\mathbf{X}) \\ \text{subject to :} \\ -\theta_l \le \tilde{\theta}_q^{(k)} \le \theta_u, \\ -\kappa_{\max} \le \tilde{\kappa}_q^{(k)} \le \kappa_{\max}, \\ \left(k = 1, \ldots, n_q\right)\end{gathered} \tag{2.3.11}$$

where n_n is the number of nodes in the finite element mesh, θ_l and θ_u are the lower and upper limits on the fiber angle orientation, respectively, $\mathbf{X}$ is the design variables vector and

$$f_i(\mathbf{X}) = \begin{cases} \left(\tilde{V}_{j_1} - V_{j_1}^*\right) & i = 2j-1 \quad (j \in \mathbb{N}) \\ \left(\tilde{V}_{j_3} - V_{j_3}^*\right) & i = 2j \end{cases} \tag{2.3.12}$$

with $V_{j_1}^*$ and $V_{j_3}^*$ being the desired optimal lamination parameters for each node j.

For the solution of the minimization problem, the nonlinear code DFNLP (Schittkowski, 2005) is implemented. The authors mention that the problem possibly has local minima, which is taken into account in the solution process.

The obtained compliance for the balanced and symmetric cantilever plate (Figure 2.11), after adding the manufacturing constraint, differs from the target optimal one obtained with the lamination parameters. However, adding manufacturing capability or increasing the number of layers used, the fiber path results get closer to the initial compliance results.

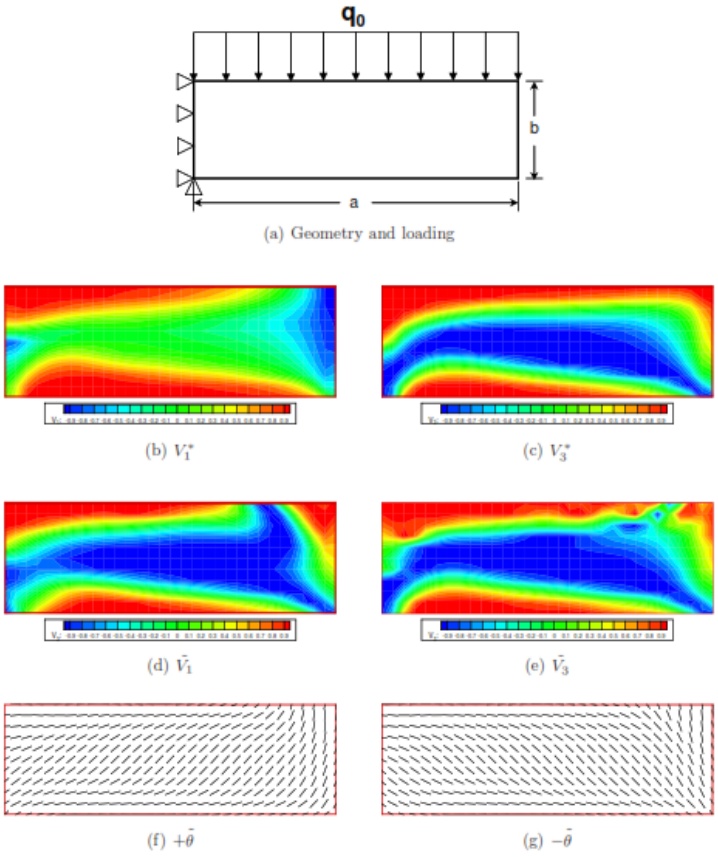


Figure 2.11: (a) Cantilever beam geometry (b),(c) Optimal lamination parameter distributions (d),(e) Computed lamination parameter distributions (f),(g) Computed fiber angles (Setoodeh, Blom, et al., 2006).

Later, Peeters, Hesse, et al. (2015) presented a multi-level approach to optimize the fiber angles, using structural approximations in order to minimize the analyses. In order to achieve global convergence, they used convex approximations. First level approximations were used for the FE response and second level approximations for the fiber angles optimization. In their work, a balanced symmetric panel with a hole was optimized for two buckling loads.

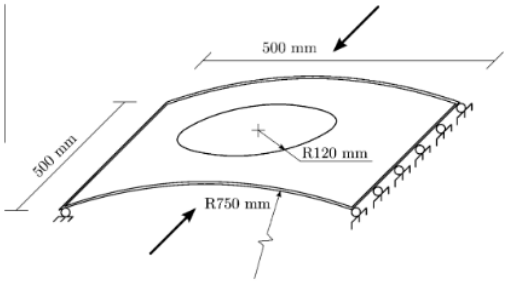
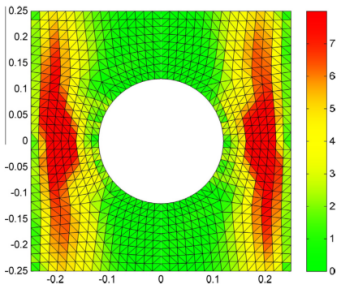
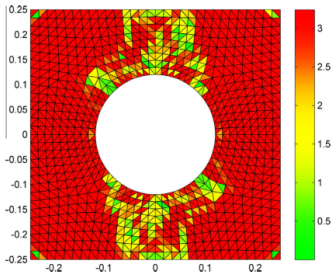


Figure 2.12: Boundary conditions for the buckling optimization of a balanced and symmetric panel with a hole (Peeters, Hesse, et al., 2015).

Regarding the manufacturing constraint, the norm of the gradient of the fiber path was constrained either locally (to prevent wrinkling) or globally (to limit gaps/overlaps). The authors observed that the global constraint could not guarantee manufacturability, since the average curvature of each ply was constrained. The maximum could be a value much higher than the average (Figure 2.13a) On the other hand, the local constraint meant a bound on the average curvature of each element, controlling the constraint in the whole ply (Figure 2.13b), but sacrificing computational time. The fiber distributions obtained with the local constraint are depicted in Figure 2.14.

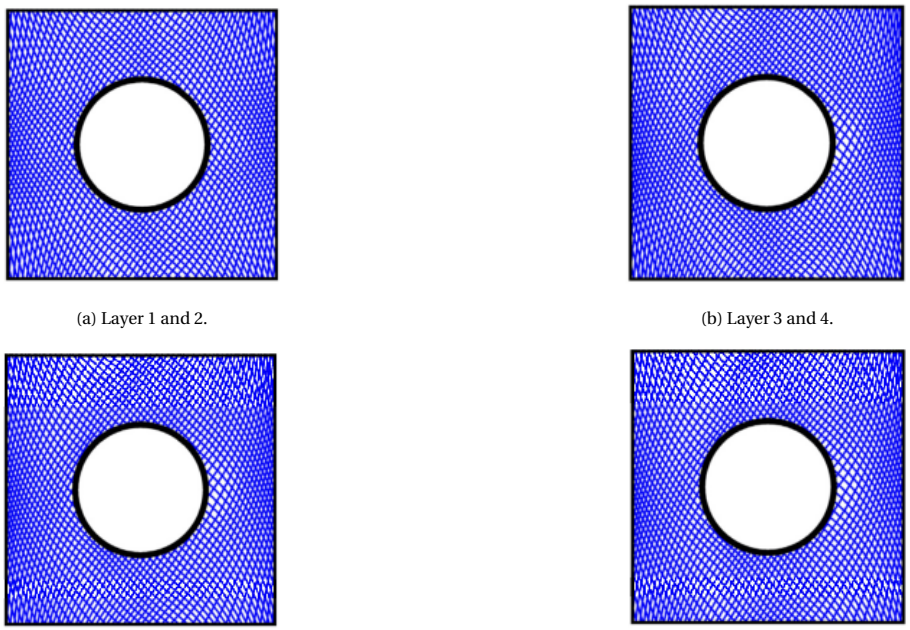


(a) Global constraint of 4.



(b) Local constraint of 3.2411.

Figure 2.13: Curvature fields (Peeters, Hesse, et al., 2015).



(a) Layer 1 and 2.

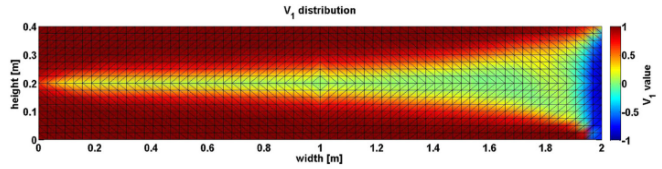
(b) Layer 3 and 4.

(c) Layer 5 and 6.

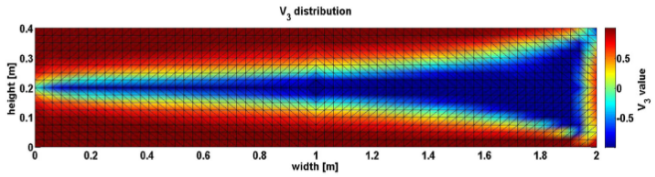
(d) Layer 7 and 8 (at symmetry plane).

Figure 2.14: Fiber paths retrieved using local steering (Peeters, Hesse, et al., 2015).

In the stacking sequence retrieval step, there have also been attempts to ensure the robustness of the laminate, by imposing common laminate design guidelines (Albazzan et al., 2019). For example, an upper limit on the ply angle difference between consecutive plies and the need for ±45° surface layers for damage tolerance have been implemented in Albazzan et al. (2018).



(a) $V_{1\mathbf{A}}$

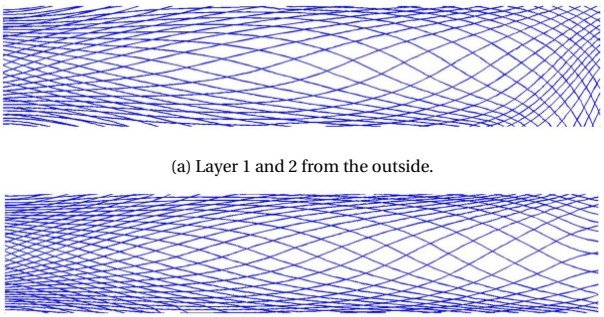


(b) $V_{3\mathbf{A}}$

Figure 2.15: Optimal lamination parameter distributions for plate under point load (Peeters et al., 2018).

Peeters et al. (2018) introduced a two-level approximation method for the compliance optimization of composite laminates, where, in the second level, the compliance is approximated in terms of either lamination parameters or fiber angles. The method of Conservative Convex Separable Approximations (CCSA) (Svanberg, 2002) is used for the optimization and the steering constraint is imposed locally as in Peeters, Hesse, et al. (2015). The optimal lamination parameters obtained for a plate, clamped on the left and under a point load applied on the bottom right side, are very similar to the results by Nagy et al. (2010), where an isogeometric approach was used. The lamination parameter distributions and the retrieved fiber angles with a steering constraint of 400 mm are shown in Figure 2.15 and Figure 2.16, respectively. The compliance obtained after the retrieval of the fiber angles is very close to the compliance of the optimal lamination parameter distribu-

tions. The developed method was also proved very computationally efficient.



(a) Layer 1 and 2 from the outside.

(b) Layer 5 and 6 from the outside.

Figure 2.16: Optimal fiber angles for plate under point load (Peeters et al., 2018).

In 2020, Hong et al. (2020) used a two-level approximation approach, as well, while imposing the curvature constraint on both step 1 and step 2 of the three-step approach. The authors named the imposition of the constraint on the fiber angles "direct method" and the imposition of the constraint on the lamination parameters "indirect method". The former one guarantees that the constraint is satisfied locally at every point. However, since it is unrelated to the optimization of step 1, it leads to a high difference in compliance between the two steps. The latter one is a strict constraint and its value cannot be decided in advance. They also tested a third method, named "hybrid", imposing the constraint both on step 1 and 2. Here, CCSA (Svanberg, 2002) was used again for the solution.

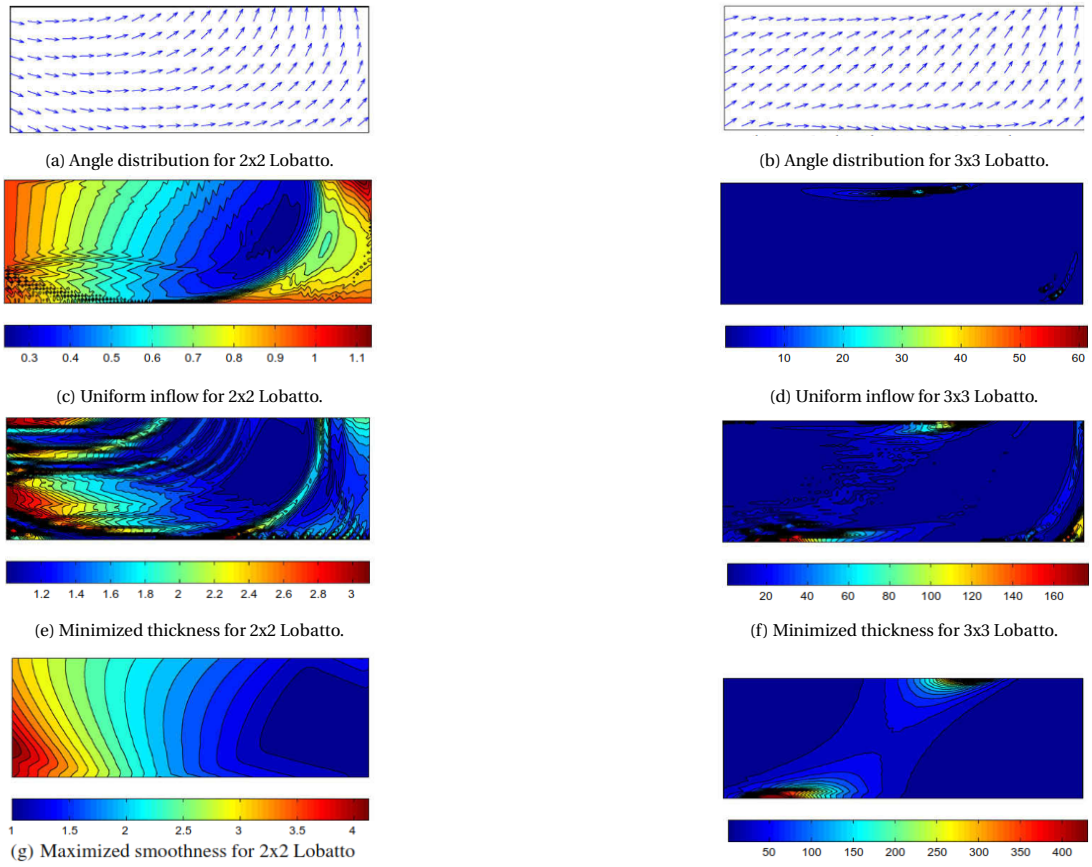
The direct method results in a considerable compliance loss between step 1 and step 2, which is expected. The distribution of lamination parameters obtained from the indirect method is almost homogeneous. Between the two steps the performance loss is low, but the steering capability of the machine is not fully exploited. The hybrid approach results in small changes to the lamination parameters, with low performance loss. Also, the obtained design is similar to the one obtained using the direct method. The hybrid approach is the one that gives the best compliance values, as well. Therefore, the authors recommend the method for a compliance minimization problem as it generates better designs and converges faster than the common in the literature direct approach.

2.3.3. Step 3 - fiber path construction

The last step in the variable stiffness design is the construction of the fiber path to be fed in the machine for manufacturing. Regarding manufacturing with fiber steering, the two popular techniques used are the so-called parallel and shifted layup. These techniques do not coincide with the optimal angle distribution retrieved from step 2, and that is where step 3 comes in.

Blom et al. (2010) used a streamline analogy to calculate the thickness build-up of the obtained distribution from step 2 and optimized the fiber courses using two optimality criteria; a maximum thickness for manufacturability and a maximum smoothness for an almost equal distribution of ply drops/overlaps. A third option suggested was to combine both of them.

The results for a 2D fiber angle distribution are shown in Figure 2.17. Second and third order Lobatto polynomials were used for the fiber angle retrieval (Setoodeh, Blom, et al., 2006). Looking at the smeared thickness distribution, there is an incompleteness in the coverage of the surface as well thickness overlaps (Figures 2.17c and 2.17d). The overlaps are extremely prominent using a third order polynomial. After thickness optimization (Figures 2.17e-2.17h) the second order polynomial was preferred to be used in conjunction with the suggested method, for better manufacturability. The suggested method can be used to see whether there is the possibility to create constant thickness laminates by eliminating any fibers that protrude out of the fiber trajectories and provides a "vivid picture" for the paths assigned for manufacturing (Albazzan et al., 2019).



(a) Angle distribution for 2x2 Lobatto.

(b) Angle distribution for 3x3 Lobatto.

(c) Uniform inflow for 2x2 Lobatto.

(d) Uniform inflow for 3x3 Lobatto.

(e) Minimized thickness for 2x2 Lobatto.

(f) Minimized thickness for 3x3 Lobatto.

(g) Maximized smoothness for 2x2 Lobatto

(g) Maximized smoothness for 2x2 Lobatto.

(h) Maximized smoothness for 3x3 Lobatto.

Figure 2.17: Fiber orientations and thickness using 2nd and 3rd order Lobatto polynomials (Blom et al., 2010).

Other works on this step include a manufacturing FE mesh, where the path is constructed with the assistance of an inserted polygon (van Tooren & Elham, 2016) and connection of the obtained paths using parabolic functions (Wu et al., 2015). The streamline analogy method of Blom et al. (2010) imposes a certain control on manufacturability issues, achieving the optimal distribution as close as possible whereas the polygon method (van Tooren & Elham, 2016) might leave gaps or overlaps that require a manual post-processing step to be dealt with (Albazzan et al., 2019).

2.4. Combination of topology and composites optimization

The existing literature that deals with topology and material optimization for laminated composite plates, where the laminate stacking sequence has to be taken into account, is limited.

Nagy et al. (2013) described a method to deal with topology, stiffness and shape optimization applied to thin-walled composite shells, using isogeometric analysis. Lamination parameters were used for the stiffness optimization and the optimum was found making use of the control point coordinates and weights as design variables. The result is obtained alternating between sizing and shape optimization in an iterative fashion. One example worth mentioning here, since it includes both thickness and lamination parameters as design variables, is the example of a square plate under uniaxial compression for the optimization of its buckling behavior (Figure 2.18).

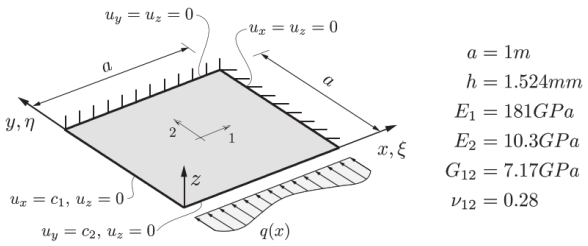


Figure 2.18: Square plate under uniaxial compression (Nagy et al., 2013).

In the optimization problem, $\boldsymbol{V}_I, \boldsymbol{W}_I, h_I$ represent the in-plane and bending lamination parameters and the thickness at each control point I, respectively. Including the thickness as a design variable improves the buckling performance. The distribution of the lamination parameters and the thickness of the plate up to a lowest thickness limit of $0.1h_0$ are depicted in Figure 2.19.

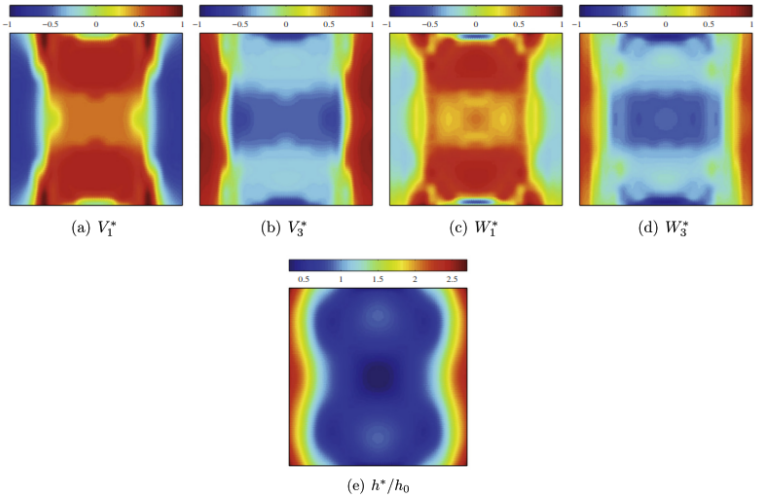


Figure 2.19: Lamination parameter and thickness distributions for a variable stiffness laminate with varying anisotropy and thickness (Nagy et al., 2013).

Later, Peeters, van Baalen, et al. (2015) combined the topology optimization method with the lamination parameter optimization for composite laminates. The optimization is followed by a post-processing procedure. The methodology is the same as the one followed in the three-step variable stiffness design (IJsselmuiden, 2011), apart from the fact that step 1 results in the optimal lamination parameter and density distributions. The second step results in the optimal angle distribution for each ply, taking into account the AFP manufacturing constraint. The third step is the fiber path construction. It has to be noted that in the first step that lamination parameters and density are optimized in a combined manner, but they are not simultaneously optimized. In order to get black and white solutions, two methods are used. The implicit method penalizes the **A** matrix using a density approach and the explicit one drives the reduction of the grey area created in the solution. Successive convex approximations are used for the optimization in this case, as well.

Regarding the post-processing, the boundary of the structure is obtained after a mesh refinement. The contour is formed then, based on a density threshold, and it is established after smoothing. The fiber angles are obtained by fitting the lamination parameter distribution while taking into account the curvature constraint. The number of plies is an input in the beginning of the optimization. For the construction of the fiber paths, the algorithm by Blom et al. (2010) is used.

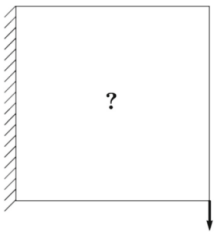
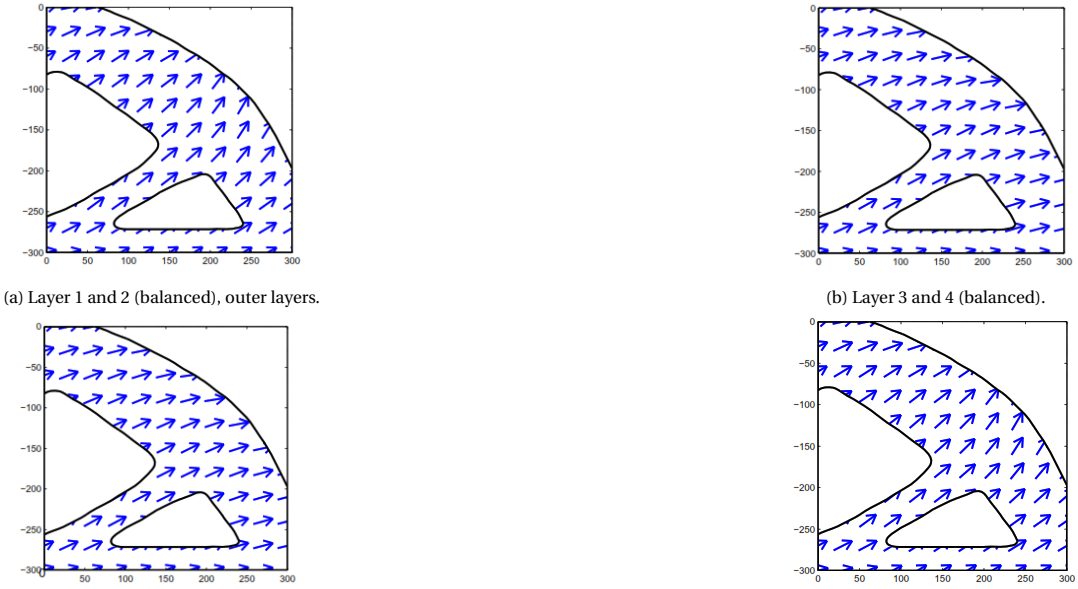


Figure 2.20: Cantilever beam example (Peeters, van Baalen, et al., 2015).

In the cantilever beam example (Figure 2.20), it was observed that the explicit penalization slightly outperforms the implicit one. The post-processing was performed on the converged case (Figure 2.21) and the fiber angles and paths were obtained subsequently (Figures 2.22 and 2.23).

(a) $V_{1\mathbf{A}}$ (b) $V_{3\mathbf{A}}$

Figure 2.21: Optimal lamination parameter distributions (Peeters, van Baalen, et al., 2015).



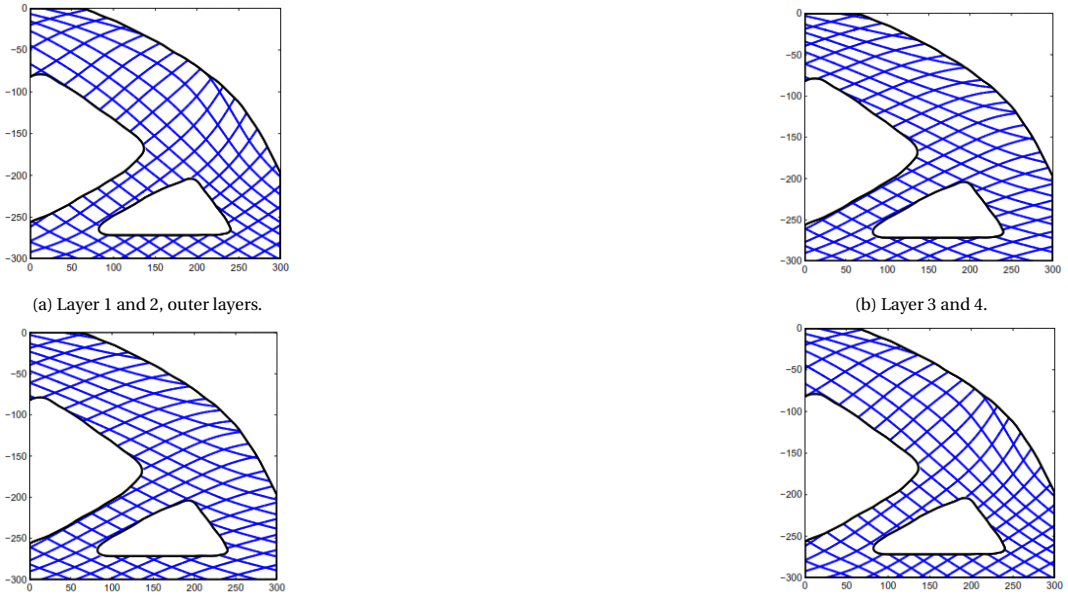
(a) Layer 1 and 2 (balanced), outer layers.

(b) Layer 3 and 4 (balanced).

(c) Layer 5 and 6 (balanced).

(d) Layer 7 and 8 (balanced), middle layers.

Figure 2.22: Optimal fiber orientations (Peeters, van Baalen, et al., 2015).



(a) Layer 1 and 2, outer layers.

(b) Layer 3 and 4.

(c) Layer 5 and 6.

(d) Layer 7 and 8, middle layers.

Figure 2.23: Optimal fiber paths (Peeters, van Baalen, et al., 2015).

2.5. Synopsis

Both topology optimization and variable stiffness design are able to provide efficient lightweight structures. Topology optimization, since its introduction, became popular primarily for isotropic materials and has been exploited so far to a much greater extent than variable stiffness design. The latter one, which has recently started gaining popularity due to the late manufacturing advances, provides the opportunity to make the best use of the capabilities of fiber reinforced composite materials.

Thus, the following questions arise; what if we combined the two optimization methods? Would the resulting structure improve in terms of performance compared to the structure obtained using only one of the aforementioned methods? This thesis' target is to couple topology optimization and variable stiffness design in a single optimization framework and discover the potential of the combination of the two. The first step of the three-step variable stiffness design will be implemented and, therefore, for the rest of the report, the laminate stiffness optimization will be referred to as variable stiffness design.

3

Methodology

This chapter presents the methodology followed in this thesis. Firstly, the theoretical background on which the linear FE problem for laminated composite structures is based, is presented in section §3.1. Subsequently, the topology optimization, variable stiffness design and staggered optimization methodologies are covered in sections §3.2, §3.3 and §3.4, respectively. This is achieved by presenting the formulation of the problems, the sensitivity analysis and the gradients to be used in the optimization process, as well as the implementation of the methodologies using the FE method.

3.1. Linear finite element analysis

3.1.1. Classical Laminated Plate Theory (CLPT)

Assumptions

The CLPT is an extension of the Classical Plate Theory for laminates. The Kirchhoff hypothesis applies, where

- Straight lines perpendicular to the midsurface before deformation remain straight after deformation.
- The transverse normals do not elongate and they rotate remaining perpendicular to the midsurface after deformation.

The above assumptions can be translated to zero normal strain $\left(\varepsilon_{zz}=0\right)$ and zero transverse shear strains $\left(\varepsilon_{xz}=0,\varepsilon_{yz}=0\right)$.

Static equilibrium and weak forms

The out-of-plane behaviour under planar loading is negligible. The equations of equilibrium for a laminated plate, taking into account only the in-plane behaviour, can be written as

$$\frac{\partial N_{xx}}{\partial x}+\frac{\partial N_{xy}}{\partial y}=0, \tag{3.1.1}$$

$$\frac{\partial N_{xy}}{\partial x}+\frac{\partial N_{yy}}{\partial y}=0, \tag{3.1.2}$$

where $\left(N_{xx},N_{yy},N_{xy}\right)$ are the in-plane force resultants.

Multiplying the two equations of equilibrium with the virtual displacements $(\delta u_0, \delta v_0)$ towards X and Y direction, respectively, and integrating over the element domain Ω^e, we get

$$0 = \int_{\Omega^e} \delta u_0 \left[-\frac{\partial N_{xx}}{\partial x} - \frac{\partial N_{xy}}{\partial y} \right] dxdy, \tag{3.1.3}$$

$$0 = \int_{\Omega^e} \delta v_0 \left[-\frac{\partial N_{xy}}{\partial x} - \frac{\partial N_{yy}}{\partial y} \right] dxdy. \tag{3.1.4}$$

In order to derive the weak forms of (3.1.3) and (3.1.4), the virtual displacements act as the weight functions. Integrating by parts, the weak forms are obtained

$$0 = \int_{\Omega^e} \left[\frac{\partial \delta u_0}{\partial x} N_{xx} + \frac{\partial \delta u_0}{\partial y} N_{xy} \right] dxdy - \oint_{\Gamma^e} \left(N_{xx} n_x + N_{xy} n_y \right) \delta u_0 ds, \tag{3.1.5}$$

$$0 = \int_{\Omega^e} \left[\frac{\partial \delta v_0}{\partial x} N_{xy} + \frac{\partial \delta v_0}{\partial y} N_{yy} \right] dxdy - \oint_{\Gamma^e} \left(N_{xy} n_x + N_{yy} n_y \right) \delta v_0 ds, \tag{3.1.6}$$

where $\left(n_x, n_y\right)$ are the direction cosines of the unit normal on the element boundary Γ^e.

SIMP model and constitutive relationships

The in-plane force resultants for a symmetric and balanced laminate, implementing the SIMP model, are given by

$$\left\{ \begin{array}{c} N_{xx} \\ N_{yy} \\ N_{xy} \end{array} \right\} = \left(\rho^e\right)^p \left[\begin{array}{ccc} A_{11} & A_{12} & 0 \\ A_{12} & A_{22} & 0 \\ 0 & 0 & A_{66} \end{array} \right] \left\{ \begin{array}{c} \varepsilon_{xx}^{(0)} \\ \varepsilon_{yy}^{(0)} \\ \gamma_{xy}^{(0)} \end{array} \right\}, \tag{3.1.7}$$

where A_{11}, A_{12}, A_{22} and A_{66} are the entries of the element in-plane stiffness matrix $\mathbf{A}^e$, ρ^e is the element density, p is the density penalty power and $\left\{\boldsymbol{\varepsilon^0}\right\}$ is the membrane strains vector.

The membrane strains $\left\{\boldsymbol{\varepsilon^0}\right\}$, assuming that the out-of-plane displacement w_0 is negligible, are

$$\left\{\varepsilon^0\right\} = \left\{ \begin{array}{c} \varepsilon_{xx}^{(0)} \\ \varepsilon_{yy}^{(0)} \\ \gamma_{xy}^{(0)} \end{array} \right\} = \left\{ \begin{array}{c} \frac{\partial u_0}{\partial x} \\ \frac{\partial v_0}{\partial y} \\ \frac{\partial u_0}{\partial y} + \frac{\partial v_0}{\partial x} \end{array} \right\}, \tag{3.1.8}$$

where $\left(u_0, v_0\right)$ are the midplane displacements towards X and Y direction, respectively.

Thus, the force resultants can be expressed as

$$N_{xx} = \left(\rho^e\right)^p \left(A_{11} \frac{\partial u_0}{\partial x} + A_{12} \frac{\partial v_0}{\partial y} \right), \tag{3.1.9}$$

$$N_{yy} = \left(\rho^e\right)^p \left(A_{12} \frac{\partial u_0}{\partial x} + A_{22} \frac{\partial v_0}{\partial y} \right), \tag{3.1.10}$$

$$N_{xy} = \left(\rho^e\right)^p A_{66} \left(\frac{\partial u_0}{\partial y} + \frac{\partial v_0}{\partial x} \right). \tag{3.1.11}$$

3.1.2. Spatial approximations

The element chosen for the FE analysis is a bilinear quadrilateral element, depicted in 3.1.

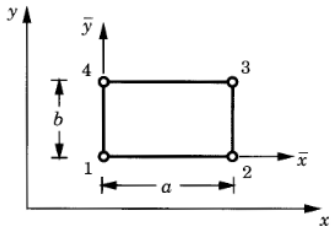


Figure 3.1: Bilinear quadrilateral element (Reddy, 2004).

The Lagrange interpolation functions for the 4 nodes of this element, in natural coordinates (ξ,η), are

$$\left\{\begin{array}{c}\psi_1\\ \psi_2\\ \psi_3\\ \psi_4\end{array}\right\}=\frac{1}{4}\left\{\begin{array}{c}(1-\xi)(1-\eta)\\ (1+\xi)(1-\eta)\\ (1+\xi)(1+\eta)\\ (1-\xi)(1+\eta)\end{array}\right\}. \tag{3.1.12}$$

The element displacements (u_0, v_0) are approximated using the Lagrange interpolation functions ψ_j over the element domain Ω^e as

$$u_0 \approx \sum_{j=1}^{4} u_j \psi_j, \tag{3.1.13}$$

$$v_0 \approx \sum_{j=1}^{4} v_j \psi_j, \tag{3.1.14}$$

where $j = 1,\ldots,4$ denotes the node of the element and $\left(u_j, v_j\right)$ are the nodal displacements towards X and Y direction, respectively.

To obtain the entries of the element in-plane stiffness matrix $\mathbf{A}^e$, we can express them as spatial approximations using the Lagrange interpolation functions ψ_j

$$A_{11} \approx \sum_{j=1}^{4} A_{11j} \psi_j, \tag{3.1.15}$$

$$A_{12} \approx \sum_{j=1}^{4} A_{12j} \psi_j, \tag{3.1.16}$$

$$A_{22} \approx \sum_{j=1}^{4} A_{22j} \psi_j, \tag{3.1.17}$$

$$A_{66} \approx \sum_{j=1}^{4} A_{66j} \psi_j, \tag{3.1.18}$$

where A_{11j}, A_{12j}, A_{22j} and A_{66j} are the nodal values of the entries A_{11}, A_{12}, A_{22} and A_{66}, respectively.

The element density values ρ^e, for topology and staggered optimization, are obtained using a linear projection function, described in subsection §3.2.3. For variable stiffness design, they are computed as spatial approximations, as discussed in subsection §3.3.4.

3.1.3. Finite element model

Element stiffness

Substituting (3.1.5) with the force resultants N_{xx} (3.1.9) and N_{xy} (3.1.11), the displacement approximations u_0 (3.1.13) and v_0 (3.1.14), and assuming that $\delta u_0 \sim \psi_i, \psi_j$, we get

$$0 = \sum_{i=1}^{4} \int_{\Omega^e} \left[\left((\rho^e)^p A_{11} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} + (\rho^e)^p A_{66} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right) u_j + \left((\rho^e)^p A_{12} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial y} + (\rho^e)^p A_{66} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial \mathrm{x}} \right) v_j \right] dxdy - \oint_{\Gamma^e} p_x \psi_i ds. \tag{3.1.19}$$

Substituting (3.1.6) with the force resultants N_{xy} (3.1.11) and N_{yy} (3.1.10), the displacement approximations u_0 (3.1.13) and v_0 (3.1.14), and assuming that $\delta v_0 \sim \psi_i, \psi_j$, we get

$$0 = \sum_{i=1}^{4} \int_{\Omega^e} \left[\left((\rho^e)^p A_{66} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial y} + (\rho^e)^p A_{12} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial x} \right) u_j + \left((\rho^e)^p A_{66} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} + (\rho^e)^p A_{22} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right) v_j \right] dxdy - \oint_{\Gamma^e} p_y \psi_i ds. \tag{3.1.20}$$

Defining

$$L_{ij} = \int_{\Omega^e} \left((\rho^e)^p A_{11} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} + (\rho^e)^p A_{66} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right) dxdy, \tag{3.1.21}$$

$$M_{ij} = \int_{\Omega^e} \left((\rho^e)^p A_{12} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial y} + (\rho^e)^p A_{66} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial x} \right) dxdy, \tag{3.1.22}$$

$$N_{ij} = \int_{\Omega^e}^{i} \left((\rho^e)^p A_{66} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial y} + (\rho^e)^p A_{12} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial x} \right) dxdy, \tag{3.1.23}$$

$$P_{ij} = \int_{\Omega^e}^{L} \left((\rho^e)^p A_{66} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} + (\rho^e)^p A_{22} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right) dxdy, \tag{3.1.24}$$

we can write Equations (3.1.5) and (3.1.6) as

$$0 = \sum_{j=1}^{4} \left(L_{ij} u_j + M_{ij} v_j \right) - \mathbf{F^1}, \tag{3.1.25}$$

$$0 = \sum_{j=1}^{4} \left(N_{ij} u_j + P_{ij} v_j \right) - \mathbf{F^2}. \tag{3.1.26}$$

The above system of linear equations can be written as

$$\mathbf{K^e u^e} = \mathbf{F^e}, \tag{3.1.27}$$

where

$$\mathbf{u^e} = \begin{Bmatrix} u_j \\ v_j \end{Bmatrix} \tag{3.1.28}$$

is the element displacement vector and

$$\mathbf{F^e} = \begin{Bmatrix} \mathbf{F^1} \\ \mathbf{F^2} \end{Bmatrix} \tag{3.1.29}$$

is the element force vector.

The element stiffness matrix can be expressed as

$$\mathbf{K^e} = \begin{bmatrix} \mathbf{K^1} \\ \mathbf{K^2} \end{bmatrix}, \tag{3.1.30}$$

where

$$\mathbf{K^1} = \begin{bmatrix} L_{ij} M_{ij} \end{bmatrix} \tag{3.1.31}$$

and

$$\mathbf{K^2} = \begin{bmatrix} N_{ij} P_{ij} \end{bmatrix}. \tag{3.1.32}$$

Gauss quadrature

For L_{ij}, substituting with (3.1.15) and (3.1.18), we have

$$\begin{aligned} L_{ij} &= \int_{\Omega^e} \left((\rho^e)^p A_{11} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} \right) dxdy + \int_{\Omega^e} \left((\rho^e)^p A_{66} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right) dxdy \\ &= \int_{\Omega^e} (\rho^e)^p \left(\sum_{j=1}^{4} A_{11j} \psi_j \right) \left(\frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} \right) dxdy + \int_{\Omega^e} (\rho^e)^p \left(\sum_{j=1}^{4} A_{66j} \psi_j \right) \left(\frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right) dxdy. \end{aligned} \tag{3.1.33}$$

Transforming the integral from Ω^e to the master element $\hat{\Omega}$, we have for L_{ij}

$$\begin{aligned} L_{ij} = &\int_{\Omega} (\rho^e)^p \left(\sum_{j=1}^{4} A_{11j} \psi_j \right) \left(J^*_{11} \frac{\partial \psi_i}{\partial \xi} + J^*_{12} \frac{\partial \psi_i}{\partial \eta} \right) \left(J^*_{11} \frac{\partial \psi_j}{\partial \xi} + J^*_{12} \frac{\partial \psi_j}{\partial \eta} \right) \mathscr{J} d\xi d\eta \\ &+ \int_{\Omega} (\rho^e)^p \left(\sum_{j=1}^{4} A_{66j} \psi_j \right) \left(J^*_{21} \frac{\partial \psi_j}{\partial \xi} + J^*_{22} \frac{\partial \psi_j}{\partial \eta} \right) \left(J^*_{21} \frac{\partial \psi_j}{\partial \xi} + J^*_{22} \frac{\partial \psi_j}{\partial \eta} \right) \mathscr{J} d\xi d\eta, \end{aligned} \tag{3.1.34}$$

where $\mathbf{J}^*$ is the inverse of the Jacobian matrix $\mathbf{J}$

$$\mathbf{J}^* = \mathbf{J^{-1}} \tag{3.1.35}$$

and $\mathscr{J}$ is the Jacobian

$$\mathscr{J} = J_{11} J_{22} - J_{12} J_{21}. \tag{3.1.36}$$

The Jacobian matrix $\mathbf{J}$ for a bilinear quadrilateral element with global nodal coordinates $\left(x_j, y_j\right)$ is given by

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \psi_1}{\partial \xi} & \frac{\partial \psi_2}{\partial \xi} & \frac{\partial \psi_3}{\partial \xi} & \frac{\partial \psi_4}{\partial \xi} \\ \frac{\partial \psi_1}{\partial \eta} & \frac{\partial \psi_2}{\partial \eta} & \frac{\partial \psi_3}{\partial \eta} & \frac{\partial \psi_4}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}. \tag{3.1.37}$$

Therefore, using the Gauss quadrature formulas for the integration over a rectangular master element $\hat{\Omega}$, L_{ij} can be expressed as

$$L_{ij} = \sum_{I=1}^{2} \sum_{J=1}^{2} F^1_{IJ}\left(\xi_I, \eta_J\right) W_I W_J + \sum_{I=1}^{2} \sum_{J=1}^{2} F^2_{IJ}\left(\xi_I, \eta_J\right) W_I W_J, \tag{3.1.38}$$

where

$$F^1_{IJ} = (\rho^e)^p \left(\sum_{j=1}^{4} A_{11j} \psi_j \right) \left(J^*_{11} \frac{\partial \psi_i}{\partial \xi} + J^*_{12} \frac{\partial \psi_i}{\partial \eta} \right) \left(J^*_{11} \frac{\partial \psi_j}{\partial \xi} + J^*_{12} \frac{\partial \psi_j}{\partial \eta} \right) \mathscr{J} \tag{3.1.39}$$

and

$$F^2_{IJ} = (\rho^e)^p \left(\sum_{j=1}^{4} A_{66j} \psi_j \right) \left(J^*_{21} \frac{\partial \psi_j}{\partial \xi} + J^*_{22} \frac{\partial \psi_j}{\partial \eta} \right) \left(J^*_{21} \frac{\partial \psi_j}{\partial \xi} + J^*_{22} \frac{\partial \psi_j}{\partial \eta} \right) \mathscr{J}. \tag{3.1.40}$$

(ξ_I, η_J) are the Gauss integration points for a surface integral, which for a master bilinear quadrilateral element (Figure 3.2) are

$$(\xi_I, \eta_J) = \left(\pm\sqrt{\frac{1}{3}}, \pm\sqrt{\frac{1}{3}}\right) \tag{3.1.41}$$

and W_I and W_J are the corresponding Gauss weights

$$W_I = W_J = 1. \tag{3.1.42}$$

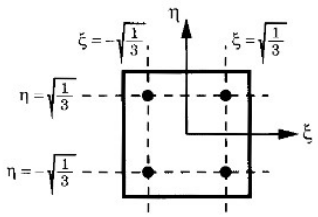


Figure 3.2: Gauss integration points locations in a master bilinear quadrilater element (Reddy, 2004).

For M_{ij}, substituting with (3.1.16) and (3.1.18), we have

$$\begin{aligned} M_{ij} &= \int_{\Omega^e} \left(\left(\rho^e\right)^p A_{12} \frac{\partial \psi_i}{\partial x}\frac{\partial \psi_j}{\partial y}\right) dxdy + \int_{\Omega^e} \left(\left(\rho^e\right)^p A_{66} \frac{\partial \psi_i}{\partial y}\frac{\partial \psi_j}{\partial x}\right) dxdy \\ &= \int_{\Omega^e} \left(\rho^e\right)^p \left(\sum_{j=1}^{4} A_{12j}\psi_j\right)\left(\frac{\partial \psi_i}{\partial x}\frac{\partial \psi_j}{\partial y}\right) dxdy + \int_{\Omega^e} \left(\rho^e\right)^p \left(\sum_{j=1}^{4} A_{66j}\psi_j\right)\left(\frac{\partial \psi_i}{\partial y}\frac{\partial \psi_j}{\partial \mathrm{x}}\right) dxdy. \end{aligned} \tag{3.1.43}$$

Transforming the integral from Ω^e to the master element $\hat{\Omega}$, we have for M_{ij}

$$\begin{aligned} M_{ij} &= \int_{\hat{\Omega}} \left(\rho^e\right)^p \left(\sum_{j=1}^{4} A_{12j}\psi_j\right)\left(J_{11}^* \frac{\partial \psi_i}{\partial \xi} + J_{12}^* \frac{\partial \psi_i}{\partial \eta}\right)\left(J_{21}^* \frac{\partial \psi_j}{\partial \xi} + J_{22}^* \frac{\partial \psi_j}{\partial \eta}\right) \mathscr{J}\, d\xi d\eta \\ &+ \int_{\hat{\Omega}} \left(\rho^e\right)^p \left(\sum_{j=1}^{4} A_{66j}\psi_j\right)\left(J_{21}^* \frac{\partial \psi_i}{\partial \xi} + J_{22}^* \frac{\partial \psi_i}{\partial \eta}\right)\left(J_{11}^* \frac{\partial \psi_j}{\partial \xi} + J_{12}^* \frac{\partial \psi_j}{\partial \eta}\right) \mathscr{J}\, d\xi d\eta. \end{aligned} \tag{3.1.44}$$

Therefore, using the Gauss quadrature formulas for the integration over a rectangular master element $\hat{\Omega}$, M_{ij} can be expressed as

$$M_{ij} = \sum_{I=1}^{2}\sum_{J=1}^{2} F_{IJ}^3\left(\xi_I, \eta_J\right) W_I W_J + \sum_{I=1}^{2}\sum_{J=1}^{2} F_{IJ}^4\left(\xi_I, \eta_J\right) W_I W_J, \tag{3.1.45}$$

where

$$F_{IJ}^3 = \left(\rho^e\right)^p \left(\sum_{j=1}^{4} A_{12j}\psi_j\right)\left(J_{11}^* \frac{\partial \psi_i}{\partial \xi} + J_{12}^* \frac{\partial \psi_i}{\partial \eta}\right)\left(J_{21}^* \frac{\partial \psi_j}{\partial \xi} + J_{22}^* \frac{\partial \psi_j}{\partial \eta}\right) \mathscr{J} \tag{3.1.46}$$

and

$$F_{IJ}^4 = \left(\rho^e\right)^p \left(\sum_{j=1}^{4} A_{66j}\psi_j\right)\left(J_{21}^* \frac{\partial \psi_i}{\partial \xi} + J_{22}^* \frac{\partial \psi_i}{\partial \eta}\right)\left(J_{11}^* \frac{\partial \psi_j}{\partial \xi} + J_{12}^* \frac{\partial \psi_j}{\partial \eta}\right) \mathscr{J}. \tag{3.1.47}$$

For N_{ij}, substituting with (3.1.18) and (3.1.16) , we have

$$\begin{aligned} N_{ij} &= \int_{\Omega^e} \left((\rho^e)^p A_{66} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial y} \right) dxdy + \int_{\Omega^e} \left((\rho^e)^p A_{12} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial x} \right) dxdy \\ &= \int_{\Omega^e} (\rho^e)^p \left(\sum_{j=1}^{4} A_{66j} \psi_j \right) \left(\frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial y} \right) dxdy + \int_{\Omega^e} (\rho^e)^p \left(\sum_{j=1}^{4} A_{12j} \psi_j \right) \left(\frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial x} \right) dxdy. \end{aligned} \tag{3.1.48}$$

Transforming the integral from Ω^e to the master element $\hat{\Omega}$, we have for N_{ij}

$$\begin{aligned} N_{ij} = &\int_{\hat{\Omega}} (\rho^e)^p \left(\sum_{j=1}^{4} A_{66j} \psi_j \right) \left(J^*_{21} \frac{\partial \psi_i}{\partial \xi} + J^*_{22} \frac{\partial \psi_i}{\partial \eta} \right) \left(J^*_{11} \frac{\partial \psi_j}{\partial \xi} + J^*_{22} \frac{\partial \psi_j}{\partial \eta} \right) \mathcal{J} d\xi d\eta \\ &+ \int_{\hat{\Omega}} (\rho^e)^p \left(\sum_{j=1}^{4} A_{12j} \psi_j \right) \left(J^*_{11} \frac{\partial \psi_i}{\partial \xi} + J^*_{12} \frac{\partial \psi_i}{\partial \eta} \right) \left(J^*_{21} \frac{\partial \psi_j}{\partial \xi} + J^*_{22} \frac{\partial \psi_j}{\partial \eta} \right) \mathcal{J} d\xi d\eta. \end{aligned} \tag{3.1.49}$$

Therefore, using the Gauss quadrature formulas for the integration over a rectangular master element $\hat{\Omega}$, N_{ij} can be expressed as

$$N_{ij} = \sum_{I=1}^{2} \sum_{J=1}^{2} F^5_{IJ}(\xi_I, \eta_J) W_l W_J + \sum_{I=1}^{2} \sum_{J=1}^{2} F^6_{IJ}(\xi_I, \eta_J) W_I W_J, \tag{3.1.50}$$

where

$$F^5_{IJ} = (\rho^e)^p \left(\sum_{j=1}^{4} A_{66j} \psi_j \right) \left(J^*_{21} \frac{\partial \psi_i}{\partial \xi} + J^*_{22} \frac{\partial \psi_i}{\partial \eta} \right) \left(J^*_{11} \frac{\partial \psi_j}{\partial \xi} + J^*_{22} \frac{\partial \psi_j}{\partial \eta} \right) \mathcal{J} \tag{3.1.51}$$

and

$$F^6_{IJ} = (\rho^e)^p \left(\sum_{j=1}^{4} A_{12j} \psi_j \right) \left(J^*_{11} \frac{\partial \psi_j}{\partial \xi} + J^*_{12} \frac{\partial \psi_j}{\partial \eta} \right) \left(J^*_{21} \frac{\partial \psi_j}{\partial \xi} + J^*_{22} \frac{\partial \psi_j}{\partial \eta} \right) \mathcal{J}. \tag{3.1.52}$$

For P_{ij}, substituting with (3.1.18) and (3.1.17) , we have

$$\begin{aligned} P_{ij} &= \int_{\Omega^e} \left((\rho^e)^p A_{66} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} \right) dxdy + \int_{\Omega^e} \left((\rho^e)^p A_{22} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right) dxdy \\ &= \int_{\Omega^e} (\rho^e)^p \left(\sum_{j=1}^{4} A_{66j} \psi_j \right) \left(\frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} \right) dxdy + \int_{\Omega^e} (\rho^e)^p \left(\sum_{j=1}^{4} A_{22j} \psi_j \right) \left(\frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right) dxdy. \end{aligned} \tag{3.1.53}$$

Transforming the integral from Ω^e to the master element $\hat{\Omega}$, we have for P_{ij}

$$\begin{aligned} P_{ij} = &\int_{\hat{\Omega}} (\rho^e)^p \left(\sum_{j=1}^{4} A_{66j} \psi_j \right) \left(J^*_{11} \frac{\partial \psi_i}{\partial \xi} + J^*_{12} \frac{\partial \psi_i}{\partial \eta} \right) \left(J^*_{11} \frac{\partial \psi_j}{\partial \xi} + J^*_{12} \frac{\partial \psi_j}{\partial \eta} \right) \mathcal{J} d\xi d\eta \\ &+ \int_{\hat{\Omega}} (\rho^e)^p \left(\sum_{j=1}^{4} A_{22j} \psi_j \right) \left(J^*_{21} \frac{\partial \psi_i}{\partial \xi} + J^*_{22} \frac{\partial \psi_i}{\partial \eta} \right) \left(J^*_{21} \frac{\partial \psi_j}{\partial \xi} + J^*_{22} \frac{\partial \psi_j}{\partial \eta} \right) \mathcal{J} d\xi d\eta. \end{aligned} \tag{3.1.54}$$

Therefore, using the Gauss quadrature formulas for the integration over a rectangular master element $\hat{\Omega}$, P_{ij} can be expressed as

$$P_{ij} = \sum_{I=1}^{2} \sum_{J=1}^{2} F^7_{IJ}(\xi_I, \eta_J) W_l W_J + \sum_{I=1}^{2} \sum_{J=1}^{2} F^8_{IJ}(\xi_I, \eta_J) W_I W_J, \tag{3.1.55}$$

where

$$F^7_{IJ} = (\rho^e)^p \left(\sum_{j=1}^{4} A_{66j} \psi_j \right) \left(J^*_{11} \frac{\partial \psi_i}{\partial \xi} + J^*_{12} \frac{\partial \psi_i}{\partial \eta} \right) \left(J^*_{11} \frac{\partial \psi_j}{\partial \xi} + J^*_{12} \frac{\partial \psi_j}{\partial \eta} \right) \mathcal{J} \tag{3.1.56}$$

and

$$F_{IJ}^{8} = \left(\rho^{e}\right)^{p} \left(\sum_{j=1}^{4} A_{22j} \psi_j \right) \left(J_{21}^{*} \frac{\partial \psi_j}{\partial \xi} + J_{22}^{*} \frac{\partial \psi_j}{\partial \eta} \right) \left(J_{21}^{*} \frac{\partial \psi_j}{\partial \xi} + J_{22}^{*} \frac{\partial \psi_j}{\partial \eta} \right) \mathscr{J}. \tag{3.1.57}$$

Tractions

In order to construct the element force vector $\mathbf{F^e}$ for the system of the linear equations (3.1.25) and (3.1.26), two boundary integrals have to be evaluated. For this reason, the coordinate along the boundary s of an element where the traction is applied, is mapped to a coordinate $-1 \le a \le 1$. Considering a constant traction p_x and/or p_y, we can, then, evaluate the entries of the element force vector $\left(f_{xj}, f_{yj}\right)$ using the Gauss quadrature for line integrals

$$f_{xj} = p_x \oint_{\Gamma^e} \psi_j ds = p_x \int_{-1}^{1} \psi_j J_s da = p_x \sum_{j=1}^{2} w_i \psi_j \left(a_i\right) J_s \left(a_i\right), \tag{3.1.58}$$

$$f_{yj} = p_y \oint_{\Gamma^e} \psi_j ds = p_y \int_{-1}^{1} \psi_j J_s da = p_y \sum_{j=1}^{2} w_i \psi_j \left(a_i\right) J_s \left(a_i\right), \tag{3.1.59}$$

where f_{xj} and f_{yj} $\left(j = 1, \ldots, 4\right)$ are the nodal forces towards X and Y direction, respectively,
J_s is the Jacobian of the side

$$J_c = \sqrt{\left(\frac{dx}{da}\right)^2 + \left(\frac{dy}{da}\right)^2}, \tag{3.1.60}$$

a_i are the Gauss integration points

$$a_i = \pm \sqrt{\frac{1}{3}} \tag{3.1.61}$$

and w_1 and w_2 are the corresponding Gauss weights

$$w_1 = w_2 = 1. \tag{3.1.62}$$

Therefore, the force vector $\mathbf{F^e}$ for a bilinear quadrilateral element is

$$\mathbf{F^e} = \left\{ \begin{array}{c} \mathbf{F^1} \\ \mathbf{F^2} \end{array} \right\} = \left\{ \begin{array}{c} \oint_{\Gamma^e} p_x \psi_j ds \\ \oint_{\Gamma^e} p_y \psi_j ds \end{array} \right\} = \left\{ \begin{array}{c} f_{x1} \\ f_{x2} \\ f_{x3} \\ f_{x4} \\ f_{y1} \\ f_{y2} \\ f_{y3} \\ f_{y4} \end{array} \right\}. \tag{3.1.63}$$

Solution of the problem, strains / stresses

After constructing the global stiffness matrix $\mathbf{K}$ and implementing the boundary conditions for the structure, the displacements $\mathbf{u}$ at the nodes are obtained solving the linear system

$$\mathbf{Ku} = \mathbf{F} \tag{3.1.64}$$

where $\mathbf{F}$ is the global force vector.

The membrane strains in each element Ω^e are acquired from equations (3.1.8) at the element centroid ($\xi = \eta = 0$). Substituting with the displacement approximations (3.1.13) and (3.1.14), we get

$$\left\{ \begin{array}{c} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{array} \right\}^e = \left\{ \begin{array}{c} \frac{\partial u_0}{\partial x} \\ \frac{\partial v_0}{\partial y} \\ \frac{\partial u_0}{\partial y} + \frac{\partial v_0}{\partial x} \end{array} \right\}^e = \sum_{j=1}^{4} \left\{ \begin{array}{c} u_j \frac{\partial \psi_j}{\partial x} \\ v_j \frac{\partial \psi_j}{\partial y} \\ u_j \frac{\partial \psi_j}{\partial y} + v_j \frac{\partial \psi_j}{\partial x} \end{array} \right\}^e. \tag{3.1.65}$$

The average stresses at each element Ω^e can be computed using the constitutive relationship

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix}^e = \hat{\mathbf{C}}^e \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{Bmatrix}^e, \tag{3.1.66}$$

where $\hat{\mathbf{C}}^\mathbf{e}$ is the SIMP element stiffness tensor, defined in subsection §3.2.1.

3.2. Topology optimization

3.2.1. Problem formulation

Suppose there is a structural domain Ω with boundary $\partial\Omega \equiv S$, with tractions applied on the boundary S_t and displacements applied on S_u. The domain is filled with a linearly elastic material.

The presence/absence of this material at each position $\mathbf{x}$ of the domain is defined by the density design parameter $\rho = \rho(\mathbf{x})$. The topology optimization problem is a minimum compliance problem, formulated as

$$\min_{\rho} F(\rho) \tag{3.2.1}$$

s.t.

$$\int_\Omega \rho dV \leq R, \tag{3.2.2}$$

$$0 \leq \rho \leq 1, \tag{3.2.3}$$

where F is the objective functional. Here, the objective functional is the structural compliance, defined as the work done on the boundary S_t by the applied tractions $\hat{\mathbf{t}}$ minus the work done on the boundary S_u by the applied displacements $\hat{\mathbf{u}}$, i.e.

$$F = \int_{S_t} \hat{\mathbf{t}} \cdot \mathbf{u} dS_t - \int_{S_u} \mathbf{t} \cdot \hat{\mathbf{u}} dS_u, \tag{3.2.4}$$

where $\mathbf{u}$ and $\mathbf{t}$ are the displacement and traction vectors, respectively.

In case the applied displacements on S_u are zero, i.e. $\mathbf{u} = \hat{\mathbf{u}} = 0$ on S_u, which applies on all the examples solved subsequently, then (3.2.4) becomes

$$F = \int_{S_t} \hat{\mathbf{t}} \cdot \mathbf{u} dS_t. \tag{3.2.5}$$

Using the work theorem, (3.2.5) is equal to double the total strain energy of the structure, which can be simply written as the strain energy

$$F = \frac{1}{2} \int_\Omega \boldsymbol{\sigma} \cdot \boldsymbol{\varepsilon} dV, \tag{3.2.6}$$

$\boldsymbol{\sigma}$, $\boldsymbol{\varepsilon}$ are the stress and the strain tensors, respectively, which are related via the constitutive relationship

$$\boldsymbol{\sigma} = \hat{\mathbf{C}} \boldsymbol{\varepsilon}, \tag{3.2.7}$$

where $\hat{\mathbf{C}}$ is the SIMP stiffness tensor, given by

$$\hat{\mathbf{C}} = \rho^p \frac{\mathbf{A}}{h}, \tag{3.2.8}$$

stemming from relationship (3.1.7).

A continuation method is used for the density penalty power, whereby p increases from a minimum value (= 1) for which the optimization problem is convex and reaches a maximum value $p_{\max}$, using a small density penalty power step size Δp at each iteration. A parametric analysis on the parameters $p_{\max}$ and Δp is covered in section §5.1.2.

The strains are related to the displacements via

$$\boldsymbol{\varepsilon} = \frac{1}{2} \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right). \tag{3.2.9}$$

Also, R is the resource constraint which represents the desired volume for the structure and (3.2.3) are the so-called box constraints.

State equations

The displacements $\mathbf{u}$ are implicit functions of the density design parameter ($\mathbf{u} = \mathbf{u}(\rho)$) and they constitute the solution of the following static boundary value problem

$$(BVP) = \begin{cases} \operatorname{div}\boldsymbol{\sigma}(\mathbf{x}) = \mathbf{0}, & \text{in } \Omega \\ \boldsymbol{\sigma}(\mathbf{x})\mathbf{n} = \hat{\mathbf{t}}(\mathbf{x}), & \text{on } \partial\Omega_t \equiv S_t \\ \mathbf{u}(\mathbf{x}) = \mathbf{0}, & \text{on } \partial\Omega_u \equiv S_u \end{cases} \tag{3.2.10}$$

where $\mathbf{n}$ is the normal vector.

Based on the above, the following relationships hold

$$\delta\hat{\mathbf{t}} = 0, \quad \text{on } \partial\Omega_t \equiv S_t, \tag{3.2.11}$$

$$\delta\mathbf{u} = 0, \quad \text{on } \partial\Omega_u \equiv S_u. \tag{3.2.12}$$

3.2.2. Sensitivity analysis

Objective functional augmented with the equilibrium constraint

Using the adjoint method for the sensitivity analysis, the objective functional augmented with the equilibrium constraint can be written as follows

$$L = \frac{1}{2}\int_\Omega \boldsymbol{\sigma}\cdot\boldsymbol{\varepsilon}\, dV + \int_\Omega \mathbf{u}^*\cdot\operatorname{div}\boldsymbol{\sigma}\, dV, \tag{3.2.13}$$

where the first integral corresponds to the strain energy of the structure and the second integral corresponds to the term used by the adjoint method incorporating the equilibrium constraint and the adjoint vector $\mathbf{u}^*$, which is the solution to the adjoint problem.

Adjoint problem

For the adjoint boundary value problem, we have the following

$$(BVP') = \begin{cases} \operatorname{div}\hat{\mathbf{C}}\boldsymbol{\varepsilon}^* = \mathbf{0}, & \text{in } \Omega \\ \boldsymbol{\varepsilon}^* = \frac{1}{2}\left(\nabla\mathbf{u}^* + \nabla\mathbf{u}^{*\,T}\right) - \boldsymbol{\varepsilon}, & \text{in } \Omega \\ \hat{\mathbf{C}}\boldsymbol{\varepsilon}^*\mathbf{n} = \mathbf{0}, & \text{on } \partial\Omega_t \equiv S_t \\ \mathbf{u}^* = \mathbf{0}, & \text{on } \partial\Omega_u \equiv S_u \end{cases} \tag{3.2.14}$$

Based on the above, the following relationships hold

$$\delta\left(\hat{\mathbf{C}}\boldsymbol{\varepsilon}^*\mathbf{n}\right) = 0, \quad \text{on } \partial\Omega_t \equiv S_t, \tag{3.2.15}$$

$$\delta\mathbf{u}^* = 0, \quad \text{on } \partial\Omega_u \equiv S_u. \tag{3.2.16}$$

Objective functional augmented with the equilibrium and design constraints

In order to include the design constraints, i.e. the box $0-\rho \leqslant 0$, $\rho-1 \leqslant 0$ and the resource constraints $\int_\Omega \rho\, dV - R \leqslant 0$ in the optimization, we have to augment L using the Lagrange multipliers $\lambda_m \geqslant 0$, $\lambda_M \geqslant 0$ and $\Lambda \geqslant 0$, constructing the Lagrangian

$$\mathscr{L}\left[\rho, \lambda_m, \lambda_M, \Lambda\right] = L[\rho] + \int_\Omega \lambda_m\left(0-\rho\right) dV + \int_\Omega \lambda_M\left(\rho-1\right) dV + \Lambda\left(\int_\Omega \rho\, dV - R\right). \tag{3.2.17}$$

Gradient

For the following derivation, we take into account the relationships that apply for the two boundary value problems (3.2.10), (3.2.14) as well as the relationships (3.2.11), (3.2.12), (3.2.15) and (3.2.16).

For the second term of (3.2.13), we have

$$\begin{aligned}\int_\Omega \mathbf{u}^* \cdot \operatorname{div}\boldsymbol{\sigma}\, dV &= \int_\Omega \operatorname{div}\left[\left(\hat{\mathbf{C}}\boldsymbol{\varepsilon}\right)\mathbf{u}^*\right] dV - \int_\Omega \hat{\boldsymbol{C}}\boldsymbol{\varepsilon}\cdot\nabla\mathbf{u}^*\, dV \\ &= \int_S \left(\hat{\mathbf{C}}\boldsymbol{\varepsilon}\right)\mathbf{u}^*\cdot\mathbf{n}dS - \int_\Omega \hat{\boldsymbol{C}}\boldsymbol{\varepsilon}\cdot\left(\nabla\mathbf{u}^* - \boldsymbol{\varepsilon}\right) dV - \int_\Omega \hat{\boldsymbol{C}}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} dV \\ &= \int_{S_t} \hat{\mathbf{t}}\cdot\mathbf{u}^* dS_t + \int_{S_u} \hat{\mathbf{t}}_{\mathbf{u}}\cdot\mathbf{u}^* dS_u - \int_\Omega \hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\left(\nabla\mathbf{u}^* - \boldsymbol{\varepsilon}\right) dV - \int_\Omega \hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} dV.\end{aligned} \tag{3.2.18}$$

The first variation of (3.2.18) with respect to ρ is

$$\begin{aligned}&\int_{S_t} \delta\hat{\mathbf{t}}\cdot\mathbf{u}^* dS_t + \int_{S_t} \hat{\mathbf{t}}\cdot\delta\mathbf{u}^* dS_t + \int_{S_u} \delta\hat{\mathbf{t}}_{\mathbf{u}}\cdot\mathbf{u}^* dS_u + \int_{S_u} \hat{\mathbf{t}}_{\mathbf{u}}\cdot\delta\mathbf{u}^* dS_u \\ &\quad - \int_\Omega \delta\hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\left(\nabla\mathbf{u}^* - \boldsymbol{\varepsilon}\right) dV - \int_\Omega \hat{\mathbf{C}}\delta\boldsymbol{\varepsilon}\cdot\left(\nabla\mathbf{u}^* - \boldsymbol{\varepsilon}\right) dV - \int_\Omega \hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\delta\boldsymbol{\varepsilon}^* dV \\ &\quad - \int_\Omega \delta\hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} dV - \int_\Omega \hat{\mathbf{C}}\delta\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} dV - \int_\Omega \hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\delta\boldsymbol{\varepsilon} dV \\ &= \int_{S_t} \hat{\mathbf{t}}\cdot\delta\mathbf{u}^* dS_t - \int_\Omega \delta\hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\left(\nabla\mathbf{u}^* - \boldsymbol{\varepsilon}\right) dV - \int_\Omega \delta\boldsymbol{\varepsilon}\cdot\hat{\mathbf{C}}\left(\nabla\mathbf{u}^* - \boldsymbol{\varepsilon}\right) dV \\ &\quad - \int_\Omega \hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\delta\boldsymbol{\varepsilon}^* dV - \int_\Omega \delta\hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} dV - \int_\Omega 2\hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\delta\boldsymbol{\varepsilon} dV.\end{aligned} \tag{3.2.19}$$

Specific terms in (3.2.19) can be further simplified. We have

$$\begin{aligned}-\int_\Omega \delta\boldsymbol{\varepsilon}\cdot\hat{\mathbf{C}}\left(\nabla\mathbf{u}^* - \boldsymbol{\varepsilon}\right) dV &= -\int_\Omega \nabla\delta\mathbf{u}\cdot\hat{\mathbf{C}}\left(\nabla\mathbf{u}^* - \boldsymbol{\varepsilon}\right) dV \\ &= -\int_\Omega \operatorname{div}\left[\hat{\mathbf{C}}\left(\nabla\mathbf{u}^* - \boldsymbol{\varepsilon}\right)\delta\mathbf{u}\right] dV + \int_\Omega \operatorname{div}\left[\hat{\mathbf{C}}\left(\nabla\mathbf{u}^* - \boldsymbol{\varepsilon}\right)\right]\cdot\delta\mathbf{u} dV \\ &= -\int_S \hat{\mathbf{C}}\left(\nabla\mathbf{u}^* - \boldsymbol{\varepsilon}\right)\delta\mathbf{u}\cdot\mathbf{n}dS \\ &= -\int_{S_u} \hat{\mathbf{C}}\left(\nabla\mathbf{u}^* - \boldsymbol{\varepsilon}\right)\mathbf{n}\cdot\delta\mathbf{u}dS_u - \int_{S_t} \hat{\mathbf{C}}\left(\nabla\mathbf{u}^* - \boldsymbol{\varepsilon}\right)\mathbf{n}\cdot\delta\mathbf{u}dS_t \\ &= -\int_{S_t} \hat{\mathbf{C}}\boldsymbol{\varepsilon}^*\mathbf{n}\cdot\delta\mathbf{u}dS_t = 0.\end{aligned} \tag{3.2.20}$$

Furthermore,

$$\begin{aligned}-2\int_\Omega \hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\delta\boldsymbol{\varepsilon} dV &= -2\int_\Omega \hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\delta\nabla\mathbf{u} dV \\ &= -2\int_\Omega \operatorname{div}\left[\left(\hat{\mathbf{C}}\boldsymbol{\varepsilon}\right)\delta\mathbf{u}\right] dV + 2\int_\Omega \operatorname{div}\left(\hat{\mathbf{C}}\boldsymbol{\varepsilon}\right)\cdot\delta\mathbf{u} dV \\ &= -2\int_S \left(\hat{\mathbf{C}}\boldsymbol{\varepsilon}\right)\delta\mathbf{u}\cdot\mathbf{n}dS + 2\int_\Omega \operatorname{div}\left(\hat{\mathbf{C}}\boldsymbol{\varepsilon}\right)\cdot\delta\mathbf{u} dV \\ &= -2\int_{S_u} \left(\hat{\mathbf{C}}\boldsymbol{\varepsilon}\right)\delta\mathbf{u}\cdot\mathbf{n}dS_u - 2\int_{S_t} \left(\hat{\mathbf{C}}\boldsymbol{\varepsilon}\right)\delta\mathbf{u}\cdot\mathbf{n}dS_t \\ &= -2\int_{S_t} \hat{\mathbf{t}}\cdot\delta\mathbf{u}dS_t.\end{aligned} \tag{3.2.21}$$

Also,

$$\begin{aligned}
-\int_\Omega \hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\delta\varepsilon^* dV &= -\int_\Omega \hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\delta\left(\nabla\mathbf{u}^*-\boldsymbol{\varepsilon}\right)dV \\
&= -\int_\Omega \hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\nabla\delta\mathbf{u}^* dV + \int_\Omega \hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\delta\boldsymbol{\varepsilon} dV \\
&= -\int_\Omega \operatorname{div}\left[\left(\hat{\mathbf{C}}\boldsymbol{\varepsilon}\right)\delta\mathbf{u}^*\right]dV + \int_\Omega \operatorname{div}\left(\hat{\mathbf{C}}\boldsymbol{\varepsilon}\right)\cdot\delta\mathbf{u}^* dV + \int_\Omega \hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\delta\boldsymbol{\varepsilon} dV \\
&= -\int_{S_u}\left(\hat{\mathbf{C}}\boldsymbol{\varepsilon}\right)\mathbf{n}\cdot\delta\mathbf{u}^* dS_u - \int_{S_t}\hat{\mathbf{t}}\cdot\delta\mathbf{u}^* dS_t + \int_{S_t}\hat{\mathbf{t}}\cdot\delta\mathbf{u} dS_t \\
&= -\int_{S_t}\hat{\mathbf{t}}\cdot\delta\mathbf{u}^* dS_t + \int_{S_t}\hat{\mathbf{t}}\cdot\delta\mathbf{u} dS_t.
\end{aligned} \tag{3.2.22}$$

Thus, (3.2.19), making use of (3.2.20), (3.2.21) and (3.2.22), becomes

$$\begin{aligned}
&\int_{S_t}\hat{\mathbf{t}}\cdot\delta\mathbf{u}^* dS_t - \int_\Omega \delta\hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\left(\nabla\mathbf{u}^*-\boldsymbol{\varepsilon}\right)dV - \int_{S_t}\hat{\mathbf{t}}\cdot\delta\mathbf{u}^* dS_t + \int_{S_t}\hat{\mathbf{t}}\cdot\delta\mathbf{u} dS_t - \int_\Omega \delta\hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} dV - 2\int_{S_t}\hat{\mathbf{t}}\cdot\delta\mathbf{u} dS_t \\
&\qquad = -\int_\Omega \delta\hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\left(\nabla\mathbf{u}^*-\boldsymbol{\varepsilon}\right)dV - \int_\Omega \delta\hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} dV - \int_{S_t}\hat{\mathbf{t}}\cdot\delta\mathbf{u} dS_t.
\end{aligned} \tag{3.2.23}$$

The first variation of the first term of (3.2.13) with respect to ρ, making use of 3.2.21, is

$$\begin{aligned}
\frac{1}{2}\int_\Omega \delta\boldsymbol{\sigma}\cdot\boldsymbol{\varepsilon} dV + \frac{1}{2}\int_\Omega \boldsymbol{\sigma}\cdot\delta\boldsymbol{\varepsilon} dV &= \frac{1}{2}\int_\Omega \delta\hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} dV + \frac{1}{2}\int_\Omega \hat{\mathbf{C}}\delta\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} dV + \frac{1}{2}\int_\Omega \hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\delta\boldsymbol{\varepsilon} dV \\
&= \frac{1}{2}\int_\Omega \delta\hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} dV + \int_\Omega \hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\delta\boldsymbol{\varepsilon} dV \\
&= \frac{1}{2}\int_\Omega \delta\hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} dV + \int_{S_t}\hat{\mathbf{t}}\cdot\delta\mathbf{u} dS_t.
\end{aligned} \tag{3.2.24}$$

Therefore, the first variation of the objective functional L, making use of (3.2.23) and (3.2.24), is

$$\begin{aligned}
\delta L[\rho;\delta\rho] &= -\int_\Omega \delta\hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\left(\nabla\mathbf{u}^*-\boldsymbol{\varepsilon}\right)dV - \int_\Omega \delta\hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} dV - \int_{S_t}\hat{\mathbf{t}}\cdot\delta\mathbf{u} dS_t + \frac{1}{2}\int_\Omega \delta\hat{\mathbf{C}}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} dV + \int_{S_t}\hat{\mathbf{t}}\cdot\delta\mathbf{u} dS_t \\
&= \frac{1}{2}\int_\Omega \frac{\partial\hat{\mathbf{C}}}{\partial\rho}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon}\delta\rho dV - \int_\Omega \frac{\partial\hat{\mathbf{C}}}{\partial\rho}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon}\delta\rho dV - \int_\Omega \frac{\partial\hat{\mathbf{C}}}{\partial\rho}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon}^*\delta\rho dV \\
&= -\frac{1}{2}\int_\Omega \frac{\partial\hat{\mathbf{C}}}{\partial\rho}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon}\delta\rho dV - \int_\Omega \frac{\partial\hat{\mathbf{C}}}{\partial\rho}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon}^*\delta\rho dV.
\end{aligned} \tag{3.2.25}$$

From (3.2.8), we have for $\frac{\partial\hat{\mathbf{C}}}{\partial\rho}$

$$\frac{\partial\hat{\mathbf{C}}}{\partial\rho} = \frac{1}{h}p\rho^{p-1}\mathbf{A}. \tag{3.2.26}$$

Substituting (3.2.26) in (3.2.25), we get

$$\begin{aligned}
\delta L[\rho;\delta\rho] &= -\frac{1}{2}\int_\Omega \frac{1}{h}p\rho^{p-1}\mathbf{A}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon}\delta\rho dV - \int_\Omega \frac{1}{h}p\rho^{p-1}\mathbf{A}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon}^*\delta\rho dV \\
&= \int_\Omega\left(-\frac{1}{2}\frac{1}{h}p\rho^{p-1}\mathbf{A}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} - \frac{1}{h}p\rho^{p-1}\mathbf{A}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon}^*\right)\delta\rho dV.
\end{aligned} \tag{3.2.27}$$

Moreover, since the problem is self-adjoint (i.e. $\mathbf{u}^* = \mathbf{u}$), we have that

$$\boldsymbol{\varepsilon}^* = \frac{1}{2}\left(\nabla\mathbf{u}^* + \nabla\mathbf{u}^{*T}\right) - \boldsymbol{\varepsilon} = \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon} = \mathbf{0} \tag{3.2.28}$$

and (3.2.27) becomes

$$\delta L[\rho;\delta\rho] = \int_\Omega\left(-\frac{1}{2}\frac{1}{h}p\rho^{p-1}\mathbf{A}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon}\right)\delta\rho dV. \tag{3.2.29}$$

The first variation of the augmented objective functional (3.2.17) with respect to ρ is

$$\delta\mathscr{L}\left[\rho,\lambda_m,\lambda_M,\Lambda;\delta\rho\right] = \delta L[\rho;\delta\rho] + \int_\Omega\left(-\lambda_m+\lambda_M+\Lambda\right)\delta\rho dV. \tag{3.2.30}$$

Substituting with (3.2.29), we have

$$\delta\mathscr{L}\left[\rho,\lambda_m,\lambda_M,\Lambda;\delta\rho\right]=\int_\Omega\left(-\frac{1}{2}\frac{1}{h}p\rho^{p-1}\mathbf{A}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon}-\lambda_m+\lambda_M+\Lambda\right)\delta\rho dV, \tag{3.2.31}$$

where the gradient of the augmented objective functional is

$$G=-\frac{1}{2}\frac{1}{h}p\rho^{p-1}\mathbf{A}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon}-\lambda_m+\lambda_M+\Lambda. \tag{3.2.32}$$

The gradient can be alternatively expressed as

$$G=-p\frac{1}{\rho}w-\lambda_m+\lambda_M+\Lambda, \tag{3.2.33}$$

where w is the strain energy density of the structure, given by

$$w=\frac{1}{2}\boldsymbol{\sigma}\cdot\boldsymbol{\varepsilon}=\frac{1}{2}\frac{1}{h}\rho^p\mathbf{A}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon}. \tag{3.2.34}$$

3.2.3. FE discretization

Linear projection scheme

The stresses at each element Ω^e are calculated using the SIMP stiffness tensor from (3.1.66). The SIMP element stiffness tensor, is given by

$$\hat{\mathbf{C}}^e=\left(\rho^e\right)^p\frac{\mathbf{A}^e}{h}, \tag{3.2.35}$$

where the entries of $\mathbf{A}^e$ are defined by (3.1.15), (3.1.16),(3.1.17) and (3.1.18).

The element density ρ^e is defined through a projection scheme (Guest et al., 2004), which serves as a density filter. This projection scheme is introduced to obtain the desirable black and white solution and, at the same time, impose a minimum allowable member radius $r_{\min}$ in the final design. Towards that purpose, a linear projection function is used, which calculates each element density ρ^e based on the nodal density values ρ_j that lie within a distance r from the element centroid position $\bar{\mathbf{x}}^e$ lower or equal to $r_{\min}$. This means that a node at a position $\mathbf{x}$ lies inside the eligible region Ω_w^e, when

$$\mathbf{x}\in\Omega_w^e \quad \text{if } r\equiv\left\|\mathbf{x}-\bar{\mathbf{x}}^e\right\|\leqslant r_{\min} \tag{3.2.36}$$

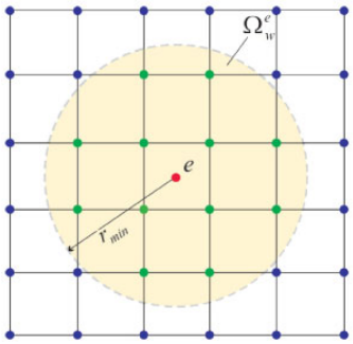


Figure 3.3: Eligible nodes located within $r_{\min}$ from $\bar{\mathbf{x}}^e$ (Guest et al., 2004)

The linear projection function is given by

$$w\left(\mathbf{x}-\bar{\mathbf{x}}^e\right)=\left\{\begin{array}{cc}\frac{r_{\min}-r}{r_{\min}} & \text{if } \mathbf{x}\in\Omega_w^e\\ 0 & \text{otherwise}\end{array}\right\} \tag{3.2.37}$$

and serves as a weight function, since every nodal value located within $r_{\min}$ from $\bar{\mathbf{x}}^e$ is attributed a certain weight, decreasing in a linear manner from 1 at the location of $\bar{\mathbf{x}}^e$ to 0 at a distance $r_{\min}$ from $\bar{\mathbf{x}}^e$.

Therefore, ρ^e is given by

$$\rho^{\mathrm{e}} = \frac{\sum_{j\in S_e} \rho_j\, w\left(\mathbf{x}_j - \bar{\mathbf{x}}^e\right)}{\sum_{j\in S_e} w\left(\mathbf{x}_j - \bar{\mathbf{x}}^e\right)}, \tag{3.2.38}$$

where
ρ_j are the nodal density values
S_e is the set of nodes located within $r_{\min}$ from $\bar{\mathbf{x}}^e$ and
$\mathbf{x}_j$ are the nodal positions.

A parametric analysis investigating the effect of different values of $r_{\min}$ is presented in subsection §5.1.2.

Design update

The design update is performed using the steepest descent method, which determines the descent direction. The aim of the method is to find the direction towards which, the objective function F, at a specific iteration of the optimization, decreases in the fastest manner locally (Arora, 2012).

At a specific iteration $k+1$, the design parameter ρ_j^{k+1} of a node j is changed based on the value ρ_j^k of the previous iteration k, following the scheme

$$\rho_j^{k+1} = \rho_j^k - a_{TO}\left(G_j + \Lambda\right), \tag{3.2.39}$$

where
a_{TO} is the step size along the search direction of the optimization and
G_j is the nodal value of the gradient.

In order to sustain continuity of the nodal values, the nodal value of the gradient G_j is expressed as a function of the weighted average of the strain energy density values of the neighboring elements

$$G_j = -p\frac{1}{\rho_j}\frac{1}{2}\left(\sum_{e=1}^{m} A^e \boldsymbol{\sigma}^{\boldsymbol{e}}\boldsymbol{\varepsilon}^{\boldsymbol{e}}\right)\frac{1}{\sum_{e=1}^{m} A^e} = -p\frac{1}{\rho_j} w_j, \tag{3.2.40}$$

where
m is the number of the neighboring elements of node j,
$\boldsymbol{\sigma}^{\boldsymbol{e}}$ is the element stress vector,
$\boldsymbol{\varepsilon}^{\boldsymbol{e}}$ is the element strain vector and
A^e is the element area.

3.3. Variable stiffness design

3.3.1. Lamination parameters

The stiffness of a laminate can be expressed in terms of twelve lamination parameters in total (Tsai & Hahn, 1980), which are repeated here for consistency

$$\begin{aligned}
\left(V_{1\mathbf{A}}, V_{2\mathbf{A}}, V_{3\mathbf{A}}, V_{4\mathbf{A}}\right) &= \int_{-\frac{1}{2}}^{\frac{1}{2}} (\cos 2\theta, \sin 2\theta, \cos 4\theta, \sin 4\theta)\, d\bar{z}, \\
\left(V_{1\mathbf{B}}, V_{2\mathbf{B}}, V_{3\mathbf{B}}, V_{4\mathbf{B}}\right) &= 4\int_{-\frac{1}{2}}^{\frac{1}{2}} \bar{z}(\cos 2\theta, \sin 2\theta, \cos 4\theta, \sin 4\theta)\, d\bar{z}, \\
\left(V_{1\mathbf{D}}, V_{2\mathbf{D}}, V_{3\mathbf{D}}, V_{4\mathbf{D}}\right) &= 12\int_{-\frac{1}{2}}^{\frac{1}{2}} \bar{z}^2(\cos 2\theta, \sin 2\theta, \cos 4\theta, \sin 4\theta)\, d\bar{z}.
\end{aligned} \tag{3.3.1}$$

In this thesis, assuming balanced and symmetric laminates under planar loading, we have

$$V_{2\mathbf{A}} = V_{4\mathbf{A}} = 0, \tag{3.3.2}$$

and

$$V_{i\mathbf{B}} = V_{i\mathbf{D}} = 0 \ \ (i = 1,2,3,4). \tag{3.3.3}$$

The in-plane **A** matrix of the **ABD** matrix varies linearly with respect to the lamination parameters

$$\mathbf{A} = h\left(\mathbf{\Gamma}_0 + \mathbf{\Gamma}_1 V_{1\mathbf{A}} + \mathbf{\Gamma}_3 V_{3\mathbf{A}}\right), \tag{3.3.4}$$

where $\mathbf{\Gamma}_0$, $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_3$ are given by

$$\begin{aligned}
\mathbf{\Gamma}_0 &= \begin{bmatrix} U_1 & U_4 & 0 \\ U_4 & U_1 & 0 \\ 0 & 0 & U_5 \end{bmatrix}, \\
\mathbf{\Gamma}_1 &= \begin{bmatrix} U_2 & 0 & 0 \\ 0 & -U_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\
\mathbf{\Gamma}_3 &= \begin{bmatrix} U_3 & -U_3 & 0 \\ -U_3 & U_3 & 0 \\ 0 & 0 & -U_3 \end{bmatrix},
\end{aligned} \tag{3.3.5}$$

where U_i's are the material invariant properties which are independent of the orientation of the fibers and defined as

$$\begin{aligned}
U_1 &= \left(3Q_{11} + 3Q_{22} + 2Q_{12} + 4Q_{66}\right)/8, \\
U_2 &= \left(Q_{11} - Q_{22}\right)/2, \\
U_3 &= \left(Q_{11} + Q_{22} - 2Q_{12} - 4Q_{66}\right)/8, \\
U_4 &= \left(Q_{11} + Q_{22} + 6Q_{12} - 4Q_{66}\right)/8, \\
U_5 &= \left(Q_{11} + Q_{22} - 2Q_{12} + 4Q_{66}\right)/8,
\end{aligned} \tag{3.3.6}$$

where Q_{ij}'s are the components of the reduced stiffness matrix, given by

$$\begin{aligned}
Q_{11} &= \frac{E_1}{1 - \nu_{12}\nu_{21}}, \\
Q_{12} &= \frac{\nu_{12} E_2}{1 - \nu_{12}\nu_{21}}, \\
Q_{22} &= \frac{E_2}{1 - \nu_{12}\nu_{21}}, \\
Q_{66} &= G_{12},
\end{aligned} \tag{3.3.7}$$

where
E_i's are the Young's moduli in material direction i
ν_{ij}'s are the Poisson's ratios $i - j$ and
G_{12} is the shear modulus in the 1-2 plane.

3.3.2. Problem formulation

Suppose there is a structural domain Ω with boundary $\partial\Omega \equiv S$, with tractions applied on the boundary S_t and displacements applied on S_u. The domain is filled with a linearly elastic material.

The stiffness of this material at each position $\mathbf{x}$ of the domain is defined by the lamination parameters $V_{i\mathbf{A}} = V_{i\mathbf{A}}(\mathbf{x})$ $(i = 1,3)$. The variable stiffness design problem is a minimum compliance problem, formulated as

$$\min_{V_{i\mathbf{A}}} F(V_{i\mathbf{A}}) \tag{3.3.8}$$

s.t.

$$\begin{aligned}
V_{3\mathbf{A}} &\geq 2V_{1\mathbf{A}}^2 - 1, \\
-1 \leq V_{i\mathbf{A}} &\leq 1 \ \ (i = 1,3),
\end{aligned} \tag{3.3.9}$$

where F is the compliance, that is the strain energy of the structure, given by (3.2.6).

3.3.3. Gradients

The variable stiffness design is a self-adjoint problem and the sensitivity analysis is performed in the same manner as it is performed for the topology optimization problem. Making use of the same state equations and following the same procedure for the sensitivity analysis, we find that the explicit derivatives of the SIMP stiffness tensor $\hat{\mathbf{C}}$ with respect to the lamination parameters $V_{1\mathbf{A}}$ and $V_{3\mathbf{A}}$ are given by

$$\begin{aligned} \frac{d\hat{\mathbf{C}}}{dV_{1\mathbf{A}}} &= \frac{1}{h}\rho^p \frac{d\mathbf{A}}{dV_{1\mathbf{A}}} = \rho^p \boldsymbol{\Gamma}_{\mathbf{1}}, \\ \frac{d\hat{\mathbf{C}}}{dV_{3\mathbf{A}}} &= \frac{1}{h}\rho^p \frac{d\mathbf{A}}{dV_{3\mathbf{A}}} = \rho^p \boldsymbol{\Gamma}_{\mathbf{3}} \end{aligned} \tag{3.3.10}$$

and, therefore, the gradients to be used in the optimization are, respectively

$$\begin{aligned} G_1 &= -\tfrac{1}{2}\frac{d\hat{\mathbf{C}}}{dV_{1\mathbf{A}}}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} = -\tfrac{1}{2}\rho^p \boldsymbol{\Gamma}_1 \boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon}, \\ G_3 &= -\tfrac{1}{2}\frac{d\hat{\mathbf{C}}}{dV_{3\mathbf{A}}}\boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} = -\tfrac{1}{2}\rho^p \boldsymbol{\Gamma}_{\mathbf{3}} \boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon}. \end{aligned} \tag{3.3.11}$$

3.3.4. FE discretization

Spatial approximations for the element density

As already mentioned, the stresses at each element Ω^e are calculated from (3.1.66) with the SIMP element stiffness tensor given by (3.2.35).

In variable stiffness design, the element density ρ^e can be expressed, consistently with the FE method, using the Lagrange interpolation functions over the element domain Ω^e

$$\rho^e \approx \sum_{j=1}^{4} \rho_j \psi_j. \tag{3.3.12}$$

Design update

The design update is also performed using the steepest descent method.

At a specific iteration $k+1$, the design parameters $V_{1\mathbf{A}j}^{k+1}$ and $V_{3\mathbf{A}j}^{k+1}$ of a node j are changed based on the values $V_{1\mathbf{A}j}^{k}$ and $V_{3\mathbf{A}j}^{k}$ of the previous iteration k, following the scheme

$$V_{1\mathbf{A}j}^{k+1} = V_{1j}^{k} - a_{VS,1} G_{1j}, \tag{3.3.13}$$

$$V_{3\mathbf{A}j}^{k+1} = V_{3j}^{k} - a_{VS,3} G_{3j}, \tag{3.3.14}$$

where
$a_{VS,i}$ $(i = 1,3)$ is the step size along the search direction of the optimization
G_{ij} $(i = 1,3)$ is the nodal value of the gradient.

In a similar manner to the computation of the gradient nodal values for topology optimization, G_{1j} and G_{3j} for each node j can be expressed as

$$G_{1j} = -\frac{1}{2}\rho_j^p \left(\sum_{e=1}^{m} A_e \boldsymbol{\Gamma}_1 \boldsymbol{\varepsilon}^{\boldsymbol{e}} \boldsymbol{\varepsilon}^{\boldsymbol{e}} \right) \frac{1}{\sum_{e=1}^{m} A_e} \tag{3.3.15}$$

$$G_{3j} = -\frac{1}{2}\rho_j^p \left(\sum_{e=1}^{m} A_e \boldsymbol{\Gamma}_3 \boldsymbol{\varepsilon}^{\boldsymbol{e}} \boldsymbol{\varepsilon}^{\boldsymbol{e}} \right) \frac{1}{\sum_{e=1}^{m} A_e}, \tag{3.3.16}$$

where
m is the number of the neighboring elements of node j,
$\boldsymbol{\varepsilon}^{\boldsymbol{e}}$ is the element strain vector and
A^e is the element area.

3.4. Staggered optimization

The staggered optimization is a combination of the two aforementioned methods, topology optimization and variable stiffness design. In this case, the design parameters are not simultaneously updated. This means that in one iteration, both topology optimization and variable stiffness design problems are solved in a sequential manner. First, the densities are updated using (3.2.39), while keeping the lamination parameters constant and next, the lamination parameters are updated using (3.3.13) and (3.3.14), while keeping the densities constant. Subsequently, the algorithm proceeds to the next iteration, where both of the optimization problems are solved again. The sensitivities used for topology optimization and variable stiffness design are given by (3.2.40) and (3.3.15), (3.3.16), respectively.

A detailed description of how the staggered optimization is implemented numerically is provided in subsection §4.3.3.

4

Numerical Implementation

This chapter describes in detail how the optimization methods are implemented numerically. The whole procedure followed to obtain the desired optimized structure is implemented in a function called `main(.m)` written in MATLAB. The function is divided in four parts in total, which follow the flow depicted in Figure 4.1.

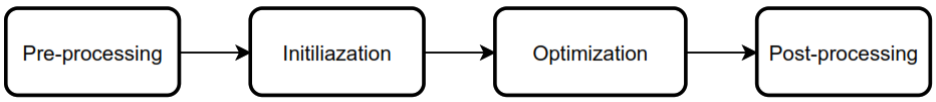


Figure 4.1: Flowchart of `main` function.

Section §4.1 explains how the geometry and the mesh of the structure to be optimized are generated and sections §4.2, §4.3 and §4.4 describe in detail the steps followed during pre-processing, initialization, optimization and post-processing, as well as the role of every function developed. Finally, section §4.5 presents the numerical implementation of the methodology for a multiple load case scenario.

4.1. Geometry and mesh generation

The software Gmsh is used for the generation of the mesh (Geuzaine & Remacle, 2009). Inside the Gmsh environment, both the geometry and the mesh can be created either using the GUI or via Gmsh's scripting language. For this thesis, a combination of both was used.

The OpenCASCADE kernel was used for the generation of the geometry, as it allows for boolean operations between elementary entities, such as lines and circles. An example geometry of a flat plate in Gmsh is depicted in Figure 4.2. In order for the geometry to be meshed, closed surfaces have to be defined first. This is achieved using the commands `Curve Loop`, which defines a closed loop of lines and `Plane Surface`, which defines the surface entity.

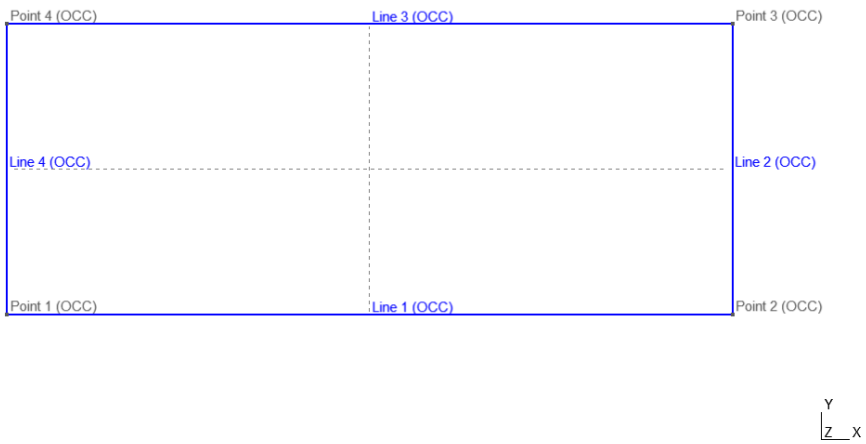


Figure 4.2: Geometry of a flat plate in Gmsh.

A 2D triangular mesh is created using the structured transfinite algorithm, as illustrated in Figure 4.3. At first, the surfaces to be meshed with this specific algorithm are selected with the command `Transfinite Surface` and the number of nodes at each edge is assigned using the command `Transfinite Curve`. The latter is combined with `Progression 1`, meaning that there is no geometrical progression of the distribution of the nodes across each edge. Next, the elements are converted to quadrilaterals using the `Recombine` command. The example quadrilateral mesh for the rectangular plate is depicted in Figure 4.4. A description of the .geo file, which contains the commands for the geometry and mesh generation in the scripting language of Gmsh, is given in Appendix A.

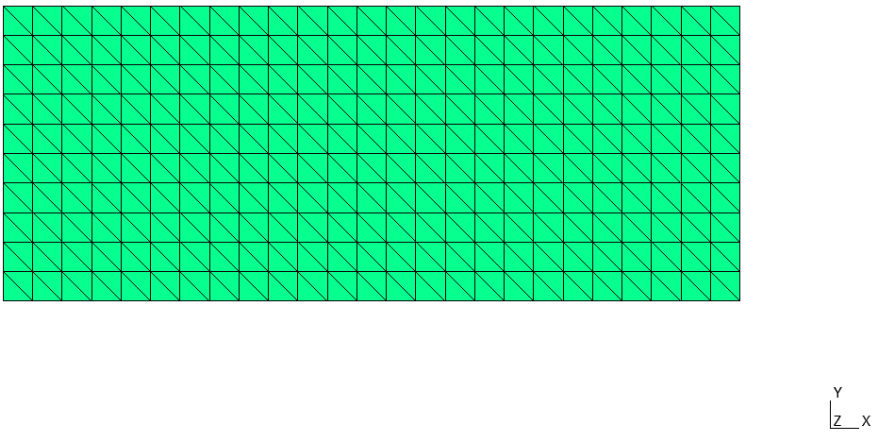


Figure 4.3: Triangular mesh of a flat plate in Gmsh.

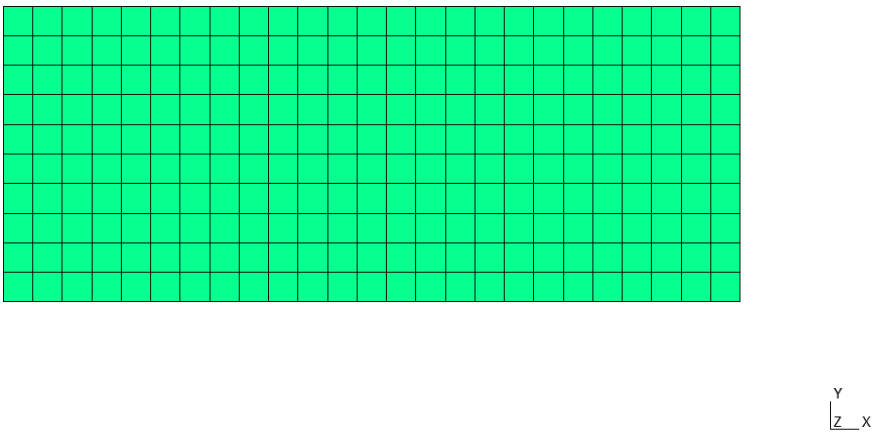


Figure 4.4: Quadrilateral mesh of a flat plate in Gmsh.

The direction of the element normals and tangents (Figure 4.5) is visually checked for consistency with the FE formulation, as implemented in MATLAB, and the mesh is exported as an .msh file, in Version 2 ASCII format. The .msh file structure is explained in Appendix B.

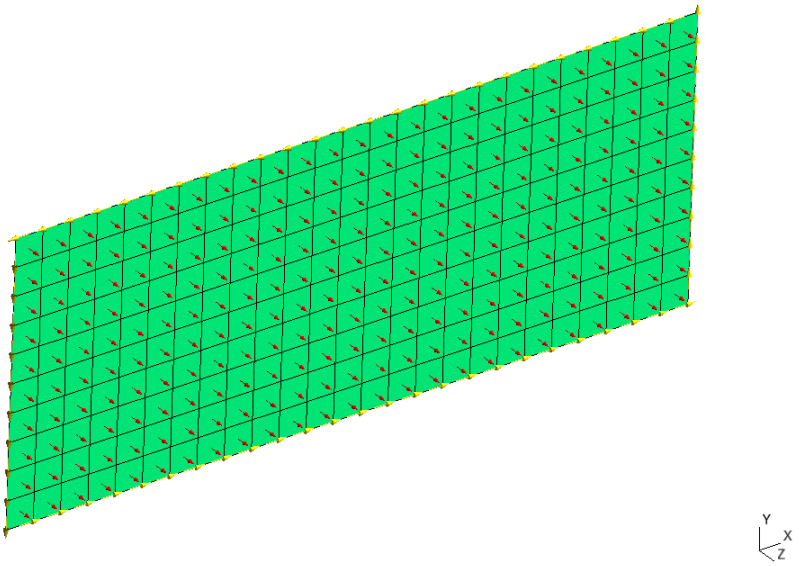


Figure 4.5: Element normals and tangents in Gmsh.

4.2. Pre-processing

The Pre-processing part comprises the definition of the input parameters for topology optimization and variable stiffness design, the mesh data input and mesh-related calculations, the implementation of the boundary conditions and the projection scheme.

The procedure followed in the Pre-processing part of `main` is depicted in the flowchart of Figure 4.6.

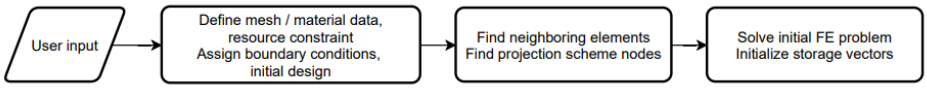


Figure 4.6: Flowchart of the Pre-processing part of `main`.

In the beginning of the Pre-processing part, the user has to specify certain parameters, analyzed in subsection §4.2.1. Next, the mesh and the material data are input in MATLAB and the resource constraint of the volume is defined. Also, the boundary conditions specified by the user are assigned and the initial design parameters are chosen, depending on the optimization method. These are achieved via the function `PreProcessingInfo`, discussed in subsection §4.2.2. The Pre-processing part continues with the identification of the neighboring elements to each node. The function used for this purpose is explained in subsection §4.2.3. Also, the eligible nodes for the projection scheme for each element are identified and their respective distance from the element centroid is calculated, with a process covered in subsection §4.2.4. The The Pre-processing part ends with solving the FE problem using the initial design and initializing the vectors to store the necessary data for each iteration, as described in subsections §4.2.5 and §4.2.6, respectively.

4.2.1. User input

The parameters which the user has to specify inside the `main` function are the following

- **optimization_opt**: The optimization method switch, which can be 1, 2 or 3. Number 1 corresponds to topology optimization only, number 2 to variable stiffness design only and number 3 to staggered optimization.
- **p_init**, **p_max** and **p_incr**: The power law penalty specifics for the density. p_init is the initial power with which the topology optimization starts, p_max is the maximum power to be reached and p_incr is the penalty power step size at each iteration.
- **PowerAddition_TO_step**: A value added to the power of the topology optimization step size, determined by user calibration.
- **PowerAddition_TO_lambda**: A value added to the power of the Lagrange multiplier Λ penalty, determined by user calibration.
- **PowerAddition_VS_step**: A value added to the power of the variable stiffness design step size, determined by user calibration.
- **rmin**: The minimum radius $r_{\min}$ for the projection scheme.
- **volume_fraction**: The desired volume fraction (resource constraint) for topology and staggered optimization.
- **iterations**: The number of topology optimization, variable stiffness design or staggered optimization iterations.

4.2.2. Pre-processing information - `PreProcessingInfo` function

Certain pre-processing information is passed into `main` via the function `PreProcessingInfo`. A flowchart describing the steps taking place inside the function is illustrated in Figure 4.7 and explained in Algorithm 1.

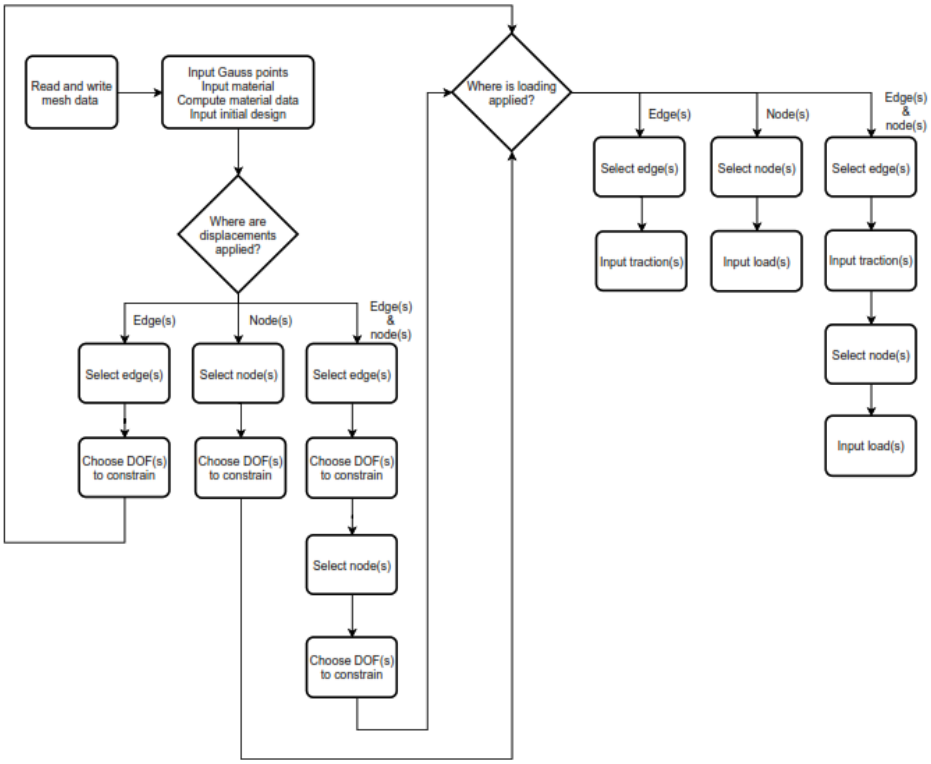


Figure 4.7: Flowchart of `PreProcessingInfo` function.

Algorithm 1 `PreProcessingInfo` function procedure.

1: The mesh data from the .msh file are read and output in MATLAB, using the function `GmshToMATLAB`.
2: The Gauss points for the bilinear quadrilateral elements and the material data are input. Also, the material invariants U_i and invariant matrices $\mathbf{\Gamma_i}$ are computed. Next, the initial design fields are input, depending on the optimization method switch selected by the user.
3: The displacement boundary condition prompt appears, via the function `BCtype`, and the user is asked to choose where the displacements are applied. If the displacements are applied on edge(s), it proceeds to step 4. If the displacements are applied on node(s), it proceeds to step 5 and if they are applied on both edge(s) and node(s), step 6 follows.
4: The user is prompted to select the edge(s) of their choice and, then, the DOF(s) to constrain for each edge, via the function `UserSelection`. The algorithm continues with step 7.
5: The user is prompted to select the node(s) of their choice and, then, the DOF(s) to constrain for each node, via the function `UserSelection`. The algorithm continues with step 7.
6: The user is prompted to select the edge(s) of their choice and, then, the DOF(s) to constrain for each edge. Next, the user is prompted to select the node(s) of their choice and the DOF(s) to constrain for each node. These are achieved via the function `UserSelection`. The algorithm continues with step 7.
7: The loading boundary condition prompt appears, via the function `BCtype`, and the user is asked to choose where the loading is applied. If the loading is applied on edge(s), it proceeds to step 8. If the loading is applied on node(s), it proceeds to step 9. If it is applied on both edge(s) and node(s), step 10 follows.
8: The user is prompted to select the edge(s) of their choice and, then, specify the traction for each edge, via the function `UserSelection`.
9: The user is prompted to select the node(s) of their choice and, then, specify the loads for each node, via the function `UserSelection`.
10: The user is prompted to select the edge(s) of their choice and, then, specify the traction for each edge. Next, the user is prompted to select the node(s) of their choice and specify the loads for each node. These are achieved via the function `UserSelection`, as well.

The input and output of `PreProcessingInfo` are described in detail in Tables 4.1 and 4.2.

Input	**Description**
gmsh_filename	Character array containing the .msh filename
optimization_opt	Optimization method switch

Table 4.1: Input of function `PreProcessingInfo`.

Output	**Description**
NumOfNodes	Number of nodes
NumOfElements	Number of elements
NodeCoords	Matrix; each row corresponds to a node, first column to its X coordinate and second column to its Y coordinate
cnc	Matrix; each row corresponds to an element, each column corresponds to one of its nodes
AllDOF	Total number of degrees of freedom
rho	Vector containing the initial density nodal values ρ_j
V1	Vector containing the initial lamination parameter nodal values $V_{1\mathbf{A}j}$
V3	Vector containing the initial lamination parameter nodal values $V_{3\mathbf{A}j}$
h	Laminate thickness
G0	Matrix $\mathbf{\Gamma_0}$
G1	Matrix $\mathbf{\Gamma_1}$
G3	Matrix $\mathbf{\Gamma_3}$
RestrainedDOF	Vector containing the restrained degrees of freedom
ForcesVector	Vector $\mathbf{F}$ containing the loading
ksi_s	Vector containing the Gauss integration points ξ_I $(I = 1,2)$
eta_s	Vector containing the Gauss integration points η_J $(J = 1,2)$

Table 4.2: Output of function `PreProcessingInfo`.

The function `PreProcessingInfo` incorporates the functions

- `GmshToMATLAB`
- `VolumeCalc`
- `BCtype`
- `UserSelection`, which includes the function `Traction`,

which will be described in detail in the following paragraphs.

The function `PreProcessingInfo` initially reads and writes mesh information to MATLAB via the function `GmshToMATLAB`.

Gmsh to MATLAB - `GmshToMATLAB` function

The input of the mesh in MATLAB is achieved using the function `GmshToMATLAB`. This function reads the mesh data from an .msh file and outputs the number of nodes, the node coordinates, the number of elements, the mesh connectivity matrix and the boundary nodes at the edges. The input and output of the function `GmshToMATLAB` are summarized in Tables 4.3 and 4.4.

Input	Description
gmsh_filename	Character array containing the .msh filename

Table 4.3: Input of function `GmshToMATLAB`.

Output	Description
NumOfNodes	See Table 4.2
NumOfElements	See Table 4.2
NodeCoordsID	Matrix; each row corresponds to a node, first column to its ID, second column to its X coordinate and third column to its Y coordinate
cncID	Matrix; each row corresponds to an element, first column to its ID and the rest to its nodes
BoundaryNodes	Matrix; each row corresponds to a boundary element edge, first column to its tag and the rest to its nodes

Table 4.4: Output of function `GmshToMATLAB`.

Firstly, the .msh input file is opened in MATLAB, using the built-in function `fopen`. Then, each line of the .msh file is read with the built-in function `fgetl` and the line where the node listing starts is sought, using the function `s_begin` (Burkardt, 2019), which compares the names of two strings. `s_begin` includes the functions `s_len_trim` and `ch_eqi` (Burkardt, 2019), which return the length of a string (excluding blank characters) and true/false if two characters are the same, respectively. The function `ch_cap` is embedded inside `ch_eqi` and returns the capital of a character.

The node listing starts with the string $Nodes. Using the built-in MATLAB function `sscanf`, the IDs as well as the node coordinates are obtained and stored in the matrix NodeCoordsID. The code identifies when the node listing is finished, denoted by the string $EndNodes.

Secondly, the line where the element listing starts is sought. This is done in a similar manner to the node listing identification. The start of the element listing is identified in the .msh file by the string $Elements. Under $Elements, not only the actual elements of the mesh are listed, but also other elementary entities, as explained in Appendix B. For this reason, the number of columns in the .msh file under $Elements is sought. If it is different than one, then there is an entity listed. If the second column is occupied by number 3, then this specific line is referring to a quadrilateral element and its connectivity. The element ID and its connectivity are then stored in the matrix cncID. If the second column is occupied by number 1, then it is referring to an edge (or, else, boundary) and its tag along with the nodes of the same are obtained and stored in the matrix BoundaryNodes. The function identifies when the node listing is finished, marked by the string $EndElements.

A mesh visualizer is embedded in the function, which can be used to plot the mesh with the respective node and element IDs, as depicted in Figures 4.8 and 4.9.

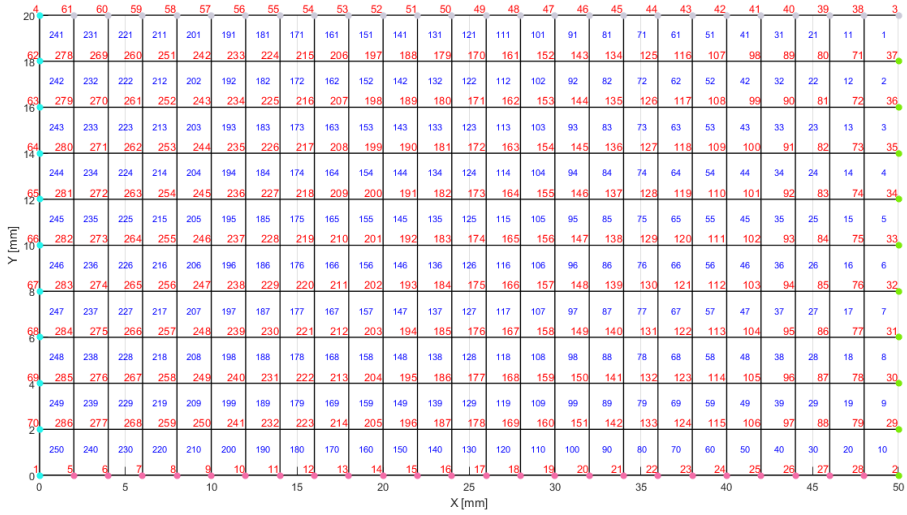


Figure 4.8: Mesh visualizer in MATLAB for a flat plate.

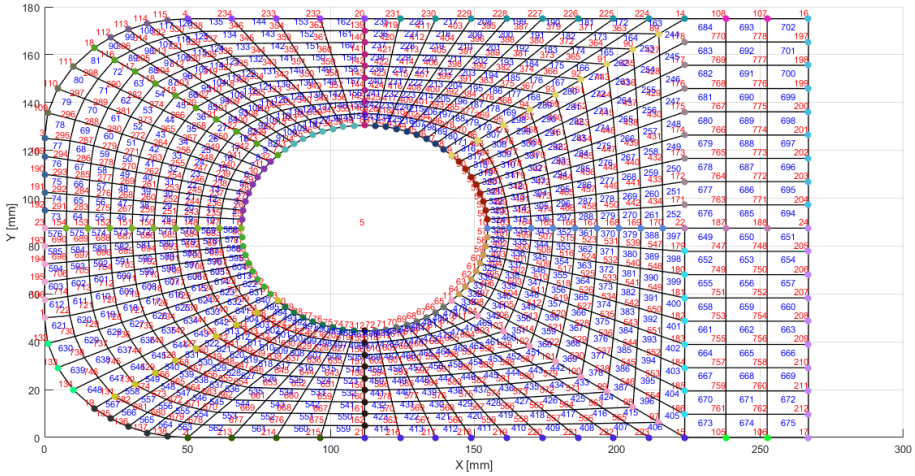


Figure 4.9: Mesh visualizer in MATLAB for a flat lug.

Subsequently, the integration points for the Gauss quadrature are given (3.1.41), the material properties are inserted, and based on these, the $U_i's$ (3.3.6) and the $\mathbf{\Gamma_i}'s$ (3.3.5) are calculated.

Next, the domain fields of the initial design are inserted. For topology optimization only, the nodal lamination parameter values are given zero values (and are kept constant throughout the optimization), whereas all of the nodal density values are assigned the value 0.5. Setting the lamination parameters to zero, the in-plane stiffness matrix $\mathbf{A}$ corresponds to a matrix of a quasi-isotropic material. For variable stiffness design only, the nodal density values are assigned the value 1 (and are kept constant throughout the optimization), whereas all of the nodal lamination parameter values are assigned the value 0.5. Last, for the staggered optimization, all the nodal values, density and lamination parameters, are assigned the value 0.5.

Then, the resource constraint is calculated. The resource constraint is simply the volume fraction, input by the user, times the volume of the full domain, i.e. when all of the density nodal values are equal to 1. The volume is calculated making use of the function `VolumeCalc`.

Volume calculation - `VolumeCalc` function

The input and output of the function `VolumeCalc` are indicated in Tables 4.5 and 4.6.

Input	Description
NumOfElements	See Table 4.2
NodeCoords	—"—
rho	—"—
cnc	—"—
ksi_s	—"—
eta_s	—"—
h	Laminate thickness

Table 4.5: Input of function `VolumeCalc`.

Output	Description
Volume	Volume of the structure

Table 4.6: Output of function `VolumeCalc`.

The volume of each element is

$$V^e = \int_{\Omega^e} \rho^e dV. \tag{4.2.1}$$

Consistently with the FE formulation and for a constant laminate thickness h, (4.2.1) can be written as

$$V^e = h \int_{\Omega^e} \left(\sum_{j=1}^{4} \rho_j^e \psi_j^e \right) dxdy, \tag{4.2.2}$$

which, transforming the integral from Ω^e to the master element $\hat{\Omega}$, becomes

$$V^e = h \int_{\hat{\Omega}} \left(\sum_{j=1}^{4} \rho_j^e \psi_j^e \right) \mathcal{J} d\xi d\eta, \tag{4.2.3}$$

and, using the Gauss quadrature formulas, can be expressed as

$$V^e = \sum_{I=1}^{2} \sum_{j=1}^{2} Z_{IJ}\left(\xi_I, \eta_J\right) W_I W_J, \tag{4.2.4}$$

where

$$Z_{IJ} = h \left(\sum_{j=1}^{4} \rho_j^e \psi_j^e \right) \mathcal{J}, \tag{4.2.5}$$

with $\left(\xi_I, \eta_J\right)$ and W_I, W_J given by (3.1.41) and (3.1.42), respectively.

The volume of the structure is calculated as the sum of the element volumes.

Afterwards, the boundary conditions are implemented. The user can specify the boundary conditions interactively, through the MATLAB GUI.

Boundary condition type - `BCtype` function

The function `BCtype` takes as input the type of the boundary condition, displacements or loading, which is indicated in the code by the numbers 1 and 2, respectively, and outputs whether the boundary condition is applied on individual nodes, edges or on a combination of the two. This is indicated via a number, 1 for nodes, 2 for edges and 3 for nodes and edges. The input and output of the function are summarized in Tables 4.7 and 4.8.

Input	Description
Type	Number indicating the boundary condition type

Table 4.7: Input of function `BCtype`.

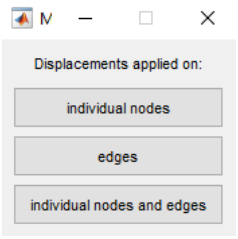
Input	Description
Option	Number indicating where the boundary condition is applied

Table 4.8: Output of function `BCtype`.

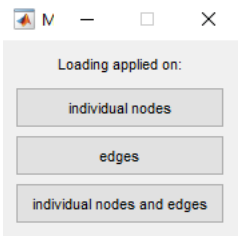
The `PreProcessingInfo` function initially asks from the user where the boundary condition of the displacements should be applied, via `BCtype`. The user is able to see a menu box, as in Figure 4.10a.

Afterwards, the user is prompted to select the nodes and/or edges of their choice. This is achieved through the function `UserSelection`, which is described in detail subsequently. In this way, the restrained degrees of freedom are stored.

Next, the `PreProcessingInfo` function asks from the user where the boundary condition of the loading should be applied, via `BCtype`, as well. The user is able to see a menu box, as in Figure 4.10b.



(a) Displacements.



(b) Loading.

Figure 4.10: Menu for the application of the boundary conditions.

The user is subsequently prompted to select the nodes and/or edges of their choice, through the function `UserSelection`. That way, the loading vector is stored.

User interaction with the GUI - `UserSelection` function

The purpose of the function `UserSelection` is to receive the type of the boundary condition and where it is applied and to return the restrained degrees of freedom, if the boundary condition type is displacements, or the loading vector, if the boundary condition is loading. The input and output of the function are summarized in Tables 4.9 and 4.10.

Input	Description
Type	See Table 4.7
Option	See Table 4.8
NumOfNodes	See Table 4.2
NumOfElements	—”—
AllDOF	—”—
cnc	—”—
BoundaryNodes	See Table 4.4
NumOfEdges	Number of edges where the displacements/loading are applied

Table 4.9: Input of function `UserSelection`.

Output	Description
DOF	Vector containing the restrained degrees of freedom/vector **F** containing the loading

Table 4.10: Output of function `UserSelection`.

For the boundary condition of displacements (Type = 1) applied on nodes (Option = 1), the user is prompted to choose the nodes on the mesh created on the GUI (Figure 4.11). For the boundary condition of displacements (Type = 1) applied on edges (Option = 2), the user is prompted to choose on how many edges the displacements will be applied and, next, any middle point of each edge, in the order they were selected (Figure 4.12). The boundary edges are recognizable in the mesh using colored dots (as illustrated in Figures 4.8 and 4.9).

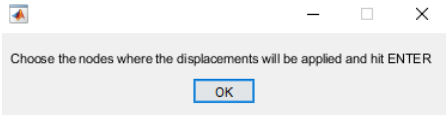
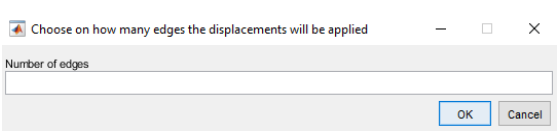
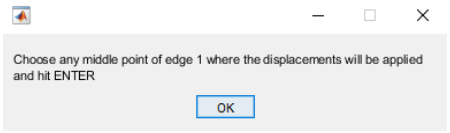


Figure 4.11: GUI prompt for node selection where the displacements will be applied.



(a) Number of edges input.



(b) Selection of a middle point for edge # 1.

Figure 4.12: GUI prompts for edge selection where the displacements will be applied.

The coordinates selected by the user are received via the built-in function `ginput` and the node IDs are recognized, based on an algorithm that seeks the node coordinates closest to the coordinates that the user clicked on the mesh. The selected nodes are stored in the vector Nodes. In case the application is on edges (Option = 2), the boundary edge where the selected node belongs to is identified. Next, the nodes that belong to that specific edge are stored in the vector Nodes.

Afterwards, the degrees of freedom to be restrained have to be identified. For this purpose, the user is prompted to choose which degrees of freedom to constrain (Figure 4.13). If X is selected, then the output is the vector that corresponds to the global DOF [2Nodes-1]. If Y is selected, then the output is the vector corresponding to the DOF [2Nodes] and if both are selected, then the output is the vector corresponding to the DOF [2Nodes-1 2Nodes].

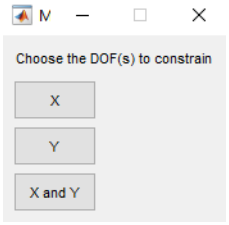


Figure 4.13: GUI prompt for selection of DOF to constrain.

The function works in a similar manner for the loading boundary condition. For loading (Type = 2) applied on nodes (Option = 1), the user is prompted to choose the nodes on the mesh created on the GUI (Figure 4.14). For loading (Type = 2) applied on edges (Option = 2), the user is prompted to select any middle point of the edge where the traction will be applied (Figure 4.15).

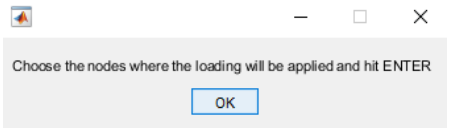
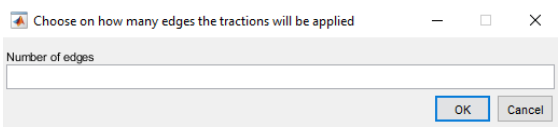
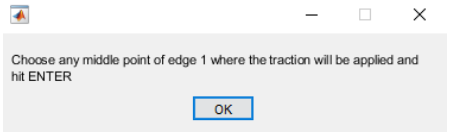


Figure 4.14: GUI prompt for node selection where the loading will be applied.



(a) Number of edges input.



(b) Selection of a middle point for edge # 1.

Figure 4.15: GUI prompts for edge selection where the tractions will be applied.

The nodes where the loading is applied are collected in the same manner, inside the vector Nodes.

In case the loading is applied on nodes (Option = 1), the user is prompted to insert the loading in direction X and/or the loading in direction Y (Figure 4.16a). Then, it is given the respective position, [2node-1] for direction X and/or [2node] for direction Y, inside the loading vector.

In case the loading is applied on edges (Option = 2), the user is prompted to insert the normal and tangent components of the traction (Figure 4.16b). The assignment of the loading vector is done via the function `Traction`, explained next.

(a) Input of node load.

(b) Input of edge traction.

Figure 4.16: GUI prompt for loading input.

If the displacements and/or loading are applied on both edge(s) and node(s) the user is prompted to select the edge(s) first and the node(s) second.

Calculation of tractions - `Traction` function

The function `Traction` returns the loading vector **F**, when the user selects to apply the loading on edges. The input and output of the function are listed in Tables 4.11 and 4.12.

Input	Description
NodeCoords	See Table 4.2
NumOfElements	—"—
cnc	—"—
AllDOF	—"—
normal	Normal traction component
tangent	Tangential traction component
EdgeNodeSequence	Vector containing the boundary (edge) loaded nodes

Table 4.11: Input of function `Traction`.

Output	Description
DOF	Vector **F** containing the loading

Table 4.12: Output of function `Traction`.

First, the function searches, based on the nodes selected by `UserSelection` previously, the loaded elements and their respective boundary nodes. Afterwards, the loading of each node in the global coordinate system is calculated, based on the subsection §3.1.3 **Tractions** and assigned its position in the loading vector **F**.

4.2.3. Neighboring elements detection - `FindNeighborElements` function

The weighted average for each node, implemented to compute the gradients for topology optimization and variable stiffness design, i.e. (3.2.40), (3.3.15) and (3.3.16), requires the identification of the neighboring to that node elements. This is achieved via the function `FindNeighborElements`, with input and output as listed in Tables 4.13 and 4.14.

The function takes as input the number of nodes and the connectivity matrix. For each node, it searches the connectivity matrix and stores the elements that contain that specific node. These are the neighboring or, else, adjacent elements to that node. For example, the elements 107, 108, 117 and 118 in Figure 4.17 are the neighboring elements to node 167.

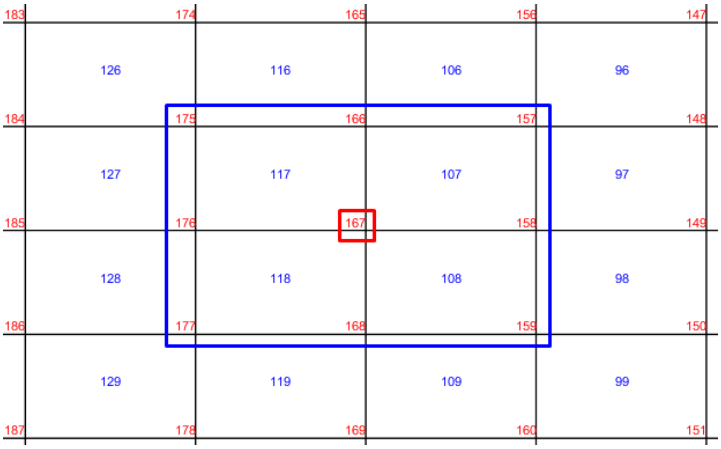


Figure 4.17: Neighboring elements of node 167.

Input	Description
NumOfNodes	See Table 4.2
cnc	—"—

Table 4.13: Input of function `FindNeighborElements`.

Output	Description
NeighboringElements	Structure; each field corresponds to a node including an array with its neighboring elements

Table 4.14: Output of function `FindNeighborElements`.

4.2.4. Projection scheme implementation - `ProjectionNodes` function

The projection scheme is implemented through the function `ProjectionNodes`. The input and output of the function ProjectionNodes are summarized in Table 4.15 and 4.16.

Initially, the coordinates of the centroid of each element are calculated. For this purpose, the element polygon is created by its four nodes using the built-in function `polyshape`. Then, the built-in function `centroid` is used to calculate the X and Y coordinates of the centroid. Next, the distance of each node to the element centroid is computed via the built-in function `pdist`. If this distance is lower or equal to $r_{\min}$, then the respective nodes are stored for the element, as well as their distance from the centroid. In case $r_{\min}$ is lower than one element's centroid distance to a node, the function assigns to the element the nodes that belong to the same.

Input	Description
rmin	Minimum radius for the projection scheme $r_{\min}$
NumOfNodes	See Table 4.2
NodeCoords	—"—
cnc	—"—

Table 4.15: Input of function `ProjectionNodes`.

Output	Description
ProjectedNodes	Structure; each row corresponds to an element and each column to a projected node of that element
Distance	Structure; each row corresponds to an element and each column to the distance of a projected node from that element centroid

Table 4.16: Output of function `ProjectionNodes`.

4.2.5. FE problem -`FESolve/FESolveVS` functions

The function used for the solution of the FE problem depends on the optimization method switch. For topology and staggered optimization, the function `FEsolve` is used, whereas for variable stiffness design alone, the function `FEsolveVS` is used. The main code initially solves the FE problem outside the optimization loop to obtain the starting values for the stress and strain fields and the compliance.

Firstly, the global stiffness matrix has to be computed. For this purpose, every element's stiffness matrix is calculated using the function `ElementStiffness` for topology and staggered optimization or the function `ElementStiffnessVS` for variable stiffness design.

These last two functions compute the element stiffness matrix as described in subsection §3.1.3. The difference of the two functions lies in the way the ρ^e is calculated for each approach. For topology and staggered optimization, the projection scheme is used for the calculation of ρ^e, given by (3.2.38), whereas for variable stiffness design, ρ^e is is calculated as a spatial approximation of the nodal values, given by (3.3.12).

Element density values from projection scheme - `ProjectionRho` function

The projection scheme is implemented via the function `ProjectionRho`, the input and output of which are summarized in Tables 4.17 and 4.18. The function calculates for every element the distance r of its projected nodes to the centroid and, subsequently, computes the element density values from (3.2.38).

Input	Description
ProjectedNodes	Cell array; each cell contains the projected nodes of an element
Distance	Cell array; each cell contains the distance of the projected nodes of an element from its centroid
rho	See Table 4.2
NumOfElements	—"—
rmin	See Table 4.15

Table 4.17: Input of function `ProjectionRho`.

Output	Description
rho_elements	Vector containing the element density values ρ_e

Table 4.18: Output of function `ProjectionRho`.

Subsequently, the nodal values of the in-plane stiffness matrix **A** are computed, based on the nodal lamination parameter values $V_{1\mathbf{A}j}$ and $V_{3\mathbf{A}j}$. According to (3.3.4), they are calculated as

$$\begin{aligned} A_{11j} &= h\left(\Gamma_{0,11} + \Gamma_{1,11} V_{1\mathbf{A}j} + \Gamma_{3,11} V_{3\mathbf{A}j}\right), \\ A_{12j} &= h\left(\Gamma_{0,12} + \Gamma_{1,12} V_{1\mathbf{A}j} + \Gamma_{3,12} V_{3\mathbf{A}j}\right), \\ A_{22j} &= h\left(\Gamma_{0,22} + \Gamma_{1,22} V_{1\mathbf{A}j} + \Gamma_{3,22} V_{3\mathbf{A}j}\right), \\ A_{66j} &= h\left(\Gamma_{0,33} + \Gamma_{1,33} V_{1\mathbf{A}j} + \Gamma_{3,33} V_{3\mathbf{A}j}\right), \end{aligned} \tag{4.2.6}$$

where $\Gamma_{0,ij}, \Gamma_{1,ij}, \Gamma_{3,ij}$ are the entries of $\mathbf{\Gamma_0}$, $\mathbf{\Gamma_1}$ and $\mathbf{\Gamma_3}$ matrices, respectively.

Next, the global stiffness matrix has to be computed. For each element, the density and the lamination parameter values that correspond to its nodes are identified, and, then, its stiffness matrix is constructed, based on the functions `ElementStiffness` and `ElementStiffnessVS`. Then, it is inserted in the correct position of the global stiffness matrix, that corresponds to the global degrees of freedom of the element nodes.

Element stiffness matrix - `ElementStiffness/ElementStiffnessVS` functions

The functions `ElementStiffness` and `ElementStiffnessVS` calculate the element stiffness matrix, following the procedure described in subsection §3.1.3 **Element stiffness** and **Gauss quadrature**. For topology and staggered optimization, the element density values ρ^e are an input of the function `ElementStiffness`, since they are previously calculated using `ProjectionRho`. For variable stiffness design, the element density values ρ^e are calculated within the function `ElementStiffnessVS`. The input and output of the element stiffness matrix functions are given in Tables 4.19, 4.20 and 4.21.

Input	**Description**
ElementNode	Vector containing the nodes of each element
NodeCoords	See Table 4.2
p	Density penalty power
ElementA11	Vector containing the nodal values A_{11j} of each element
ElementA12	Vector containing the nodal values A_{12j} of each element
ElementA22	Vector containing the nodal values A_{22j} of each element
ElementA66	Vector containing the nodal values A_{66j} of each element
ksi_s	See Table 4.2
eta_s	—"—
rho_e	Element density values ρ^e

Table 4.19: Input of function `ElementStiffness`.

Input	**Description**
ElementNode	Vector containing the nodes of each element
NodeCoords	See Table 4.2
p	Density penalty power
ElementA11	Vector containing the nodal values A_{11j} of each element
ElementA12	Vector containing the nodal values A_{12j} of each element
ElementA22	Vector containing the nodal values A_{22j} of each element
ElementA66	Vector containing the nodal values A_{66j} of each element
ElementRho	Vector containing the nodal values ρ_j of each element
ksi_s	See Table 4.2
eta_s	See Table 4.2

Table 4.20: Input of function `ElementStiffnessVS`.

Output	Description
Ke	Matrix; element stiffness

Table 4.21: Output of functions `ElementStiffness` and `ElementStiffnessVS`.

The functions `FEsolve` and `FEsolveVS` continue with identifying the active degrees of freedom, eliminating the restrained ones from the total degrees of freedom. Next, the linear system of equations (3.1.64) is solved for the displacements, using the built-in function `mldivide` for the active degrees of freedom, and, next, the restrained degrees of freedom are also added in the displacement matrix.

The calculation of the element strains and stresses follows subsequently, using the functions `ElementStrainsStresses` and `ElementStrainsStressesVS`.

Element strains / stresses - `ElementStrainsStresses` and `ElementStrainsStressesVS` functions

The functions `ElementStrainsStresses` and `ElementStrainsStressesVS` calculate the element strains and stresses based on (3.1.65) and (3.1.66). Their difference lies again on the element density values ρ^e used for the calculation of the SIMP element stiffness tensor $\hat{\mathbf{C}}^{\mathbf{e}}$. For topology and staggered optimization, the element density values ρ^e are an input of the function `ElementStrainsStresses`, since they are previously calculated using `ProjectionRho`. For variable stiffness design, the values are calculated within the function `ElementStrainsStressesVS`. The input and output of the functions are listed in Tables 4.22, 4.22 and 4.24.

Input	Description
ElementNode	Vector containing the nodes of each element
NodeCoords	See Table 4.2
ElementA11	Vector containing the nodal values A_{11j} of each element
ElementA12	Vector containing the nodal values A_{12j} of each element
ElementA22	Vector containing the nodal values A_{22j} of each element
ElementA66	Vector containing the nodal values A_{66j} of each element
element_disp	Vector containing the nodal displacements of the element
p	Density penalty power
h	Laminate thickness
rho_e	Element density values ρ^e

Table 4.22: Input of function `ElementStrainsStresses`.

Input	Description
ElementNode	Vector containing the nodes of each element
NodeCoords	See Table 4.2
ElementA11	Vector containing the nodal values A_{11j} of each element
ElementA12	Vector containing the nodal values A_{12j} of each element
ElementA22	Vector containing the nodal values A_{22j} of each element
ElementA66	Vector containing the nodal values A_{66j} of each element
element_disp	Vector containing the nodal displacements of the element
p	Density penalty power
h	Laminate thickness
ElementRho	Vector containing the density nodal values ρ_j of each element

Table 4.23: Input of function `ElementStrainsStressesVS`.

Output	Description
ElemStrains	Vector containing the element strains
ElemStresses	Vector containing the element stresses

Table 4.24: Output of functions `ElementStrainsStresses` and `ElementStrainsStressesVS`.

Finally, the compliance is calculated via the function `ComplianceCalc`.

Calculation of compliance - `ComplianceCalc` function

The input and output of the function are summarized in Tables 4.25 and 4.26.

Input	Description
NumOfElements	See Table 4.2
NodeCoords	—"—
cnc	—"—
Strains	Matrix containing the element strains, each row corresponds to an element
Stresses	Matrix containing the element stresses, each row corresponds to an element
h	Laminate thickness

Table 4.25: Input of function `ComplianceCalc`.

Output	Description
Compliance	Structural compliance

Table 4.26: Output of function `ComplianceCalc`.

The function computes the compliance as the strain energy of the structure, based on the element strains and stresses calculated at the element centroids. The compliance of each element is simply

$$\text{compliance}^e = \frac{1}{2}\boldsymbol{\sigma}^{\boldsymbol{e}}\boldsymbol{\varepsilon}^{\boldsymbol{e}} A^e h \tag{4.2.7}$$

and the compliance of the structure is calculated as the sum of the element compliances.

4.2.6. Initialization

In this section of the main code, the vectors that will store the compliance, the strains, the stresses, the nodal density values and the nodal lamination parameter values are initialized. Also, an initial value for the Lagrange multiplier Λ is chosen. The examples of this thesis were run with an initial Λ equal to 0.5.

4.3. Optimization

This section of the code is responsible for the optimization process that will be used. Based on the optimization switch chosen by the user, the corresponding optimization runs; topology optimization, variable stiffness design or staggered optimization.

4.3.1. Topology optimization

The topology optimization procedure is depicted in the flowchart of Figure 4.18. The outer loop designated in the flowchart is modified in the function responsible for the execution of the topology optimization, `TO`, in the way explained in Algorithm 2, step 11.

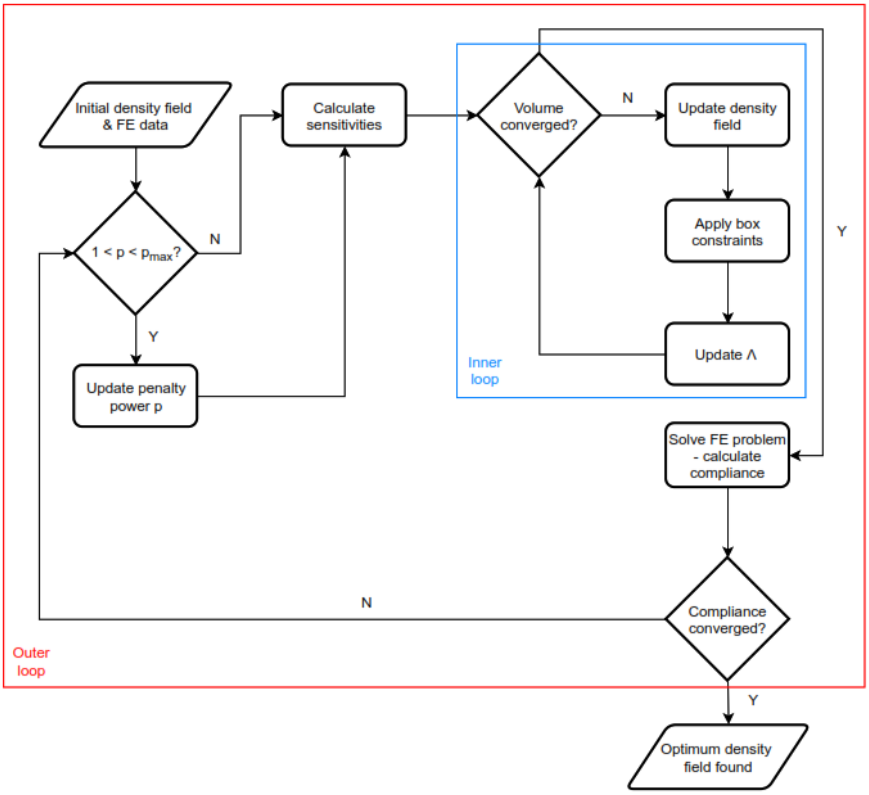


Figure 4.18: Flowchart of the topology optimization procedure.

Algorithm 2 Topology optimization procedure - `TO` function

1: The initial density field and the data from the initial solution of the FE problem (strains/stresses and compliance) are input.
2: The power law penalty power p is checked. If its value lies between 1 and $p_{\max}$, the code proceeds to step 3. Else, it proceeds to step 4.
3: The penalty power p is increased by the penalty power step size p_incr.
4: The nodal sensitivities are calculated based on the computed strains and stresses of the previously solved FE problem, using the function `SensitivitiesTO`.
5: The volume of the domain is checked for convergence. If converged, it proceeds to step 10. Else, it proceeds to step 6.
6: The density field is updated temporarily using the function `DesignUpdateTO`.
7: The box constraints are applied to the temporary density field. If ρ_j is higher than 1, it is assigned the value of 1. If it is lower than 0, it is assigned the value 0.01 to avoid singularity of the stiffness matrix.
8: The Lagrange multiplier Λ is updated using the function `LambdaUpdate`.
9: The volume of the domain is checked for convergence. If converged, the temporary density field is established and the code proceeds to step 10. Else, it goes back to step 6.
10: The FE problem is solved using the function `FESolve` based on the established density field.
11: The compliance is not checked for convergence. Instead, the solution obtained for the current iteration (density field, strains / stresses and compliance) is saved, the compliance convergence fraction is calculated and the code goes back to step 2 to repeat the process. The outer loop continues for the specified number of iterations.

The reason for the modification of the outer loop lies in the fact that the convergence tolerance is a unique value for every case solved, that depends on the optimization parameters used, the mesh density and the boundary conditions. That is why a sufficient number of outer loop iterations is prescribed by the user and the converged value is identified in the post-processing, analyzed in section §4.4.

As mentioned above, the function `TO` incorporates the following functions

- `SensitivitiesTO`
- `DesignUpdateTO`
- `LambdaUpdate`, which incorporates the function `VolumeCalc`
- `FESolve`,

which are explained subsequently, apart from the function `FESolve`, which was previously analyzed in subsection §4.2.2 **FE problem -`FESolve/FESolveVS` functions**.

The input and output of function `TO` are extensive and can be found in Tables C.1 and C.2.

Calculation of sensitivities - `SensitivitiesTO` function

The nodal sensitivities to be used for the design update are calculated using the function `SensitivitiesTO`. The input and output of the function are listed in Tables 4.27 and 4.28.

Input	Description
NodeCoords	See Table 4.2
NumOfNodes	—”—
cnc	—”—
rho	Vector containing the density nodal values ρ_j
Strains	See Table 4.25
Stresses	—”—
p	Density penalty power
NeighboringElements	Cell array; each cell contains the neighboring elements of a node

Table 4.27: Input of function `SensitivitiesTO`.

Output	Description
Gradient	Vector containing the nodal values of the gradient G_j

Table 4.28: Output of function `SensitivitiesTO`.

The sensitivities, or else the gradients to be used in the design update, are calculated using (3.2.40). For that reason, the function initially identifies the nodes of every node's neighboring elements to be used for the calculation of the element areas, computes the neighboring element values in the numerator and denominator of (3.2.40) and calculates the weighted average for the node.

Update of the design - `DesignUpdateTO` function

The design is updated, i.e. the density field is changed temporarily, using the function `DesignUpdateTO`. Its input and output is listed in Tables 4.29 and 4.30.

Input	Description
topology_loop	Number of current outer loop iteration
Gradient	See Table 4.28
lambda_temp	Lagrange multiplier Λ
rho	Vector containing the density nodal values ρ_j
PowerAddition_TO_step	Value added to the power of the topology optimization step size
step_TO	Topology optimization step size
p_init	Initial penalty power (=1)
p_max	Maximum penalty power
p_incr	Penalty power step size

Table 4.29: Input of function `DesignUpdateTO`.

Output	Description
rho_temp	Vector containing the temporary density nodal values ρ_j at current inner loop iteration
step_TO	Topology optimization step size

Table 4.30: Output of function `DesignUpdateTO`.

The function initially checks if the density penalty power p has reached its maximum value $p_{\max}$. If not, the desired order of magnitude, OrderOfMagn, for the step size is calculated as the order of magnitude of the inverse of the maximum of the gradient nodal values G_j added to the Lagrange multiplier Λ. A value is added to the result, denoted as PowerAddition_TO_step, calibrated by the user. The step size is, then

$$a_{TO} = 10^{\text{OrderOfMagn+PowerAddition\_TO\_step}} \tag{4.3.1}$$

Else, if $p_{\max}$ is reached, the optimization continues with a fixed step size. Finally, the density field is updated according to (3.2.39). The step size is also output by the function.

Update of the Lagrange multiplier - `LambdaUpdate` function

The Lagrange multiplier Λ is updated using the function `LambdaUpdate`. The input and output of the function are summarized in Tables 4.31 and 4.32.

Input	Description
resource	Resource constraint value
lambda_temp	Lagrange multiplier Λ value at current inner loop iteration
NumOfElements	See Table 4.2
NodeCoords	—"—
rho_temp	Vector containing the temporary density nodal values ρ_j at current inner loop iteration
cnc	See Table 4.2
PowerAddition_TO_lambda	Value added to the power of lambda_penalty size
ksi_s	See Table 4.2
eta_s	—"—
h	Laminate thickness

Table 4.31: Input of function `LambdaUpdate`.

Output	Description
lambda_temp	Lagrange multiplier Λ value at current inner loop iteration
volume_converge	Volume convergence percentage
volume_temp	Volume of the domain at the current inner loop iteration

Table 4.32: Output of function `LambdaUpdate`.

The function initially calculates the temporary volume of the domain, volume_temp, which changes due to the previously updated design. Next, the difference between the resource minus the temporary volume is calculated, denoted as delta_volume. Λ is calculated temporarily as

$$\Lambda_{\text{temp}} = \Lambda_{\text{temp, previous}} + \Lambda_{\text{penalty}}\texttt{delta\_volume}, \tag{4.3.2}$$

where $\Lambda_{\text{temp, previous}}$ is the Λ calculated in the previous inner loop iteration and Λ_{penalty} is given by

$$\Lambda_{\text{penalty}} = 10^{\text{OrderOfMagnL+PowerAddition\_TO\_lambda}}. \tag{4.3.3}$$

OrderOfMagnL is the order of magnitude of the inverse of delta_volume and PowerAddition_TO_lambda is an added value, determined by user calibration.

Last, the convergence fraction of the volume, volume_converge, is calculated by

$$\text{volume\_converge} = \frac{\left|\text{resource} - \text{volume\_temp}\right|}{\text{resource}}. \tag{4.3.4}$$

In this thesis, the volume convergence fraction is set to 0.01.

Before the next outer loop iteration, the compliance convergence fraction is computed as

$$\text{compliance\_change\_TO} = \frac{\left|\text{compliance} - \text{compliance\_previous}\right|}{\text{compliance}}, \tag{4.3.5}$$

where "compliance" is the compliance at the current outer loop iteration and "$\text{compliance}_{\text{previous}}$" is the compliance at the previous outer loop iteration.

4.3.2. Variable stiffness design

The variable stiffness design problem is executed using the function `VSonly`. The solution procedure is depicted in the flowchart of Figure 4.19. Here, the loop designated in the flowchart is modified in the function `VSonly`, as well, for the same reason as in topology optimization. The procedure followed is explained in Algorithm 3.

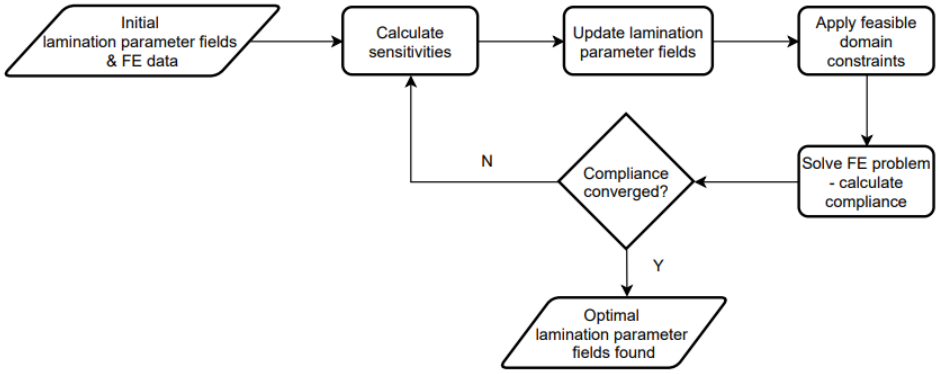


Figure 4.19: Flowchart of the variable stiffness design procedure.

Algorithm 3 Variable stiffness design procedure - `VSonly` function.

1: The initial lamination parameter fields and the data from the initial solution of the FE problem (strains / stresses and compliance) are input.
2: The nodal sensitivities are calculated based on the computed strains of the previously solved FE problem, using the function `SensitivitiesVS`.
3: The lamination parameter fields are updated temporarily using the function `DesignUpdateVS`.
4: The feasible domain constraints are applied to the temporary lamination parameter fields and the result is the established lamination parameter fields.
5: The FE problem is solved using the function `FESolveVS` based on the established lamination parameter fields.
6: The compliance is not checked for convergence. Instead, the solution obtained for the current iteration (lamination parameter fields, strains / stresses and compliance) is saved, the compliance convergence fraction is calculated and the code goes back to step 2 to repeat the process. The loop continues for the specified number of iterations.

As mentioned above, the function `VSonly` incorporates the following functions

- `SensitivitiesVS`
- `DesignUpdateVS`
- `FESolveVS`,

which are explained subsequently.

The input and output of function `VSonly` can be found in Tables C.3 and C.5.

Calculation of sensitivities - `SensitivitiesVS` function

The nodal sensitivities to be used for the design update are calculated using the function `SensitivitiesVS`. The input and output of the function are listed in Tables 4.33 and 4.34.

Input	Description
NodeCoords	See Table 4.2
NumOfNodes	—"—
cnc	—"—
rho	Vector containing the density nodal values ρ_j
Strains	See Table 4.25
p	Density penalty power
G1	See Table 4.2
G3	—"—
NeighboringElements	Cell array; each cell contains the neighboring elements of a node

Table 4.33: Input of function `SensitivitiesVS`.

Output	Description
GradientV1	Vector containing the nodal values of the gradient G_{1j}
GradientV3	Vector containing the nodal values of the gradient G_{3j}

Table 4.34: Output of function `SensitivitiesVS`.

The sensitivities, or else the gradients to be used in the design update, are calculated using (3.3.15) and (3.3.16) for each lamination parameter field. For that reason, the function initially identifies the nodes of every node's neighboring elements to be used for the calculation of the element areas, computes the neighboring element values in the numerator and denominator of (3.3.15) and (3.3.16) and calculates the weighted average for the node.

Update of the design - `DesignUpdateVS` function

The design is updated, i.e. the lamination parameter fields are changed temporarily, using the function `DesignUpdate VS`. Its input and output is listed in Tables 4.35 and 4.36.

Input	Description
varstiff_loop	Number of current variable stiffness design loop iteration
GradientV1	See Table 4.34
GradientV3	—"—
V1	Vector containing the lamination parameter nodal values $V_{1\mathbf{A}j}$
V3	Vector containing the lamination parameter nodal values $V_{1\mathbf{A}j}$
PowerAddition_VS_step	Value added to the power of the variable stiffness design step sizes
step1	Variable stiffness design step size, corresponding to the update of $V_{1\mathbf{A}}$
step2	Variable stiffness design step size, corresponding to the update of $V_{3\mathbf{A}}$

Table 4.35: Input of function `DesignUpdateVS`.

Output	Description
V1_temp	Vector containing the temporary lamination parameter nodal values $V_{1\mathbf{A}j}$ at current variable stiffness design iteration
V3_temp	Vector containing the temporary lamination parameter nodal values $V_{3\mathbf{A}j}$ at current variable stiffness design iteration
step1	Variable stiffness design step size, corresponding to the update of $V_{1\mathbf{A}}$
step2	Variable stiffness design step size, corresponding to the update of $V_{3\mathbf{A}}$

Table 4.36: Output of function `DesignUpdateVS`.

The function initially checks if the variable stiffness design iteration is the first one. If yes, then two step sizes are calculated; one for the update of $V_{1\mathbf{A}}$ and one for the update of $V_{3\mathbf{A}}$.

Similarly to the function `DesignUpdateTO`, the desired orders of magnitude for the step sizes, OrderOfMagn1 and OrderOfMagn3, respectively, are calculated as the orders of magnitude of the inverse of the maximum of the gradient nodal values G_{1j} and G_{3j}. A value is added to these results, denoted as PowerAddition_VS_step, calibrated by the user. The step sizes are, then

$$a_{VS,1} = 10^{\text{OrderOfMagn1+PowerAddition\_VS\_step}}, \tag{4.3.6}$$

$$a_{VS,3} = 10^{\text{OrderOfMagn3+PowerAddition\_VS\_step}}. \tag{4.3.7}$$

Else, if the optimization iteration is different than the first one, the process continues with fixed step sizes; the ones calculated in the first iteration.

Finally, the lamination parameter fields are updated according to (3.3.13) and (3.3.14). The step sizes are output by the function, as well.

Before the next iteration, the compliance convergence fraction is computed as

$$\text{compliance\_change\_VS} = \frac{|\text{compliance} - \text{compliance\_previous}|}{\text{compliance}}, \tag{4.3.8}$$

where "compliance" is the compliance at the current iteration and "$\text{compliance}_{\text{previous}}$" is the compliance at the previous iteration.

4.3.3. Staggered optimization

The purpose of the staggered optimization is to combine the two methods of topology optimization and variable stiffness design.

The staggered optimization is achieved by altering, at the same optimization iteration, both the density and the lamination parameter fields. The fields are not concurrently updated, but, instead, they are updated one at a time, as illustrated in the flowchart of Figure 4.20 and described in Algorithm 4. As explained before, the traditional convergence scheme depicted in the flowchart is altered and a sufficient number of iterations is used instead.

It has to be noted here that, for variable stiffness design, `VS` function is used instead of `VSonly`. As previously mentioned, the only difference between these two functions is the way the FE problem is solved. The input and output of `VS` function are listed in Tables C.4 and C.5.

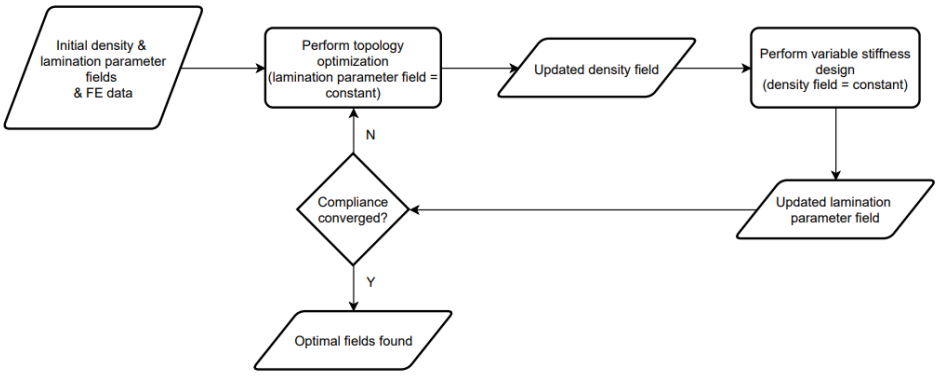


Figure 4.20: Flowchart of the staggered optimization procedure.

Algorithm 4 Staggered optimization procedure.

1: The initial density and lamination parameter fields and the data from the initial solution of the FE problem (strains / stresses and compliance) are input.
2: Topology optimization is performed using the function `TO`, while keeping the lamination parameter fields constant.
3: The compliance convergence fraction for topology optimization is calculated and the updated density field is fed into step 4.
4: Variable stiffness design is performed using the function `VS`, while keeping the density field constant.
5: The compliance convergence fraction for variable stiffness design is calculated, the updated lamination parameter fields are fed into step 2 and the loop continues for the specified number of iterations.

4.4. Post-processing

The final part of the `main` function is post-processing. This part is responsible for identifying the converged iteration, providing the compliance value, the tolerance and the computational time at convergence, via the function `PlotOptimizationResults`.

According to the optimization switch, **optimization_opt**, input by the user, the function plots the compliance values versus the number of optimization iterations, identifies the minimum compliance change tolerance and the corresponding optimization iteration and prints the converged iteration number, the compliance and the tolerance at convergence and the corresponding computational time, using the MATLAB built-in function `fprintf`. For the staggered optimization, the converged iteration is picked among the results for the topology optimization convergence fractions. This is due to the fact that it was observed that variable stiffness design converges faster than topology optimization, and, therefore, a non-converged value would be chosen for the topology optimization design otherwise.

Also, there are three functions embedded in `PlotOptimizationResults` that visualize the converged result. Which function is used depends on the optimization method. These are `PlotDensities`, `PlotLaminationParams` and `PlotCombined` for topology optimization, variable stiffness design and staggered optimization, respectively.

The input of the function `PlotOptimizationResults` is extensive and for this reason, it is listed in Table C.6.

Results visualization functions

The results visualization functions take as input the converged density and/or lamination parameter fields and plot the result. The input of the functions is listed in Table 4.37.

Input			Description
`PlotDensities`	`PlotLaminationParams`	`PlotCombined`	
cnc	cnc	cnc	See Table 4.2
NodeCoords	NodeCoords	NodeCoords	—"—
rho	-	rho	Vector containing the nodal density values ρ_j at convergence
-	V1	V1	Vector containing the nodal lamination parameter values $V_{1\mathbf{A}j}$ at convergence
-	V3	V3	Vector containing the nodal lamination parameter values $V_{3\mathbf{A}j}$ at convergence

Table 4.37: Input of results visualization functions.

The result is plotted making use of the MATLAB built-in function `patch`. This way, the elements are plotted and the colors are interpolated across the element faces based on the density and/or lamination parameter nodal values at convergence. Moreover, function `PlotCombined` uses a density threshold of 0.9. This means that the nodes that have a density value of 0.9 and above are identified, and only the elements containing these nodes are plotted.

4.5. Multiple load case implementation

In reality, most structures need to be optimized under more than one possible load cases. That is why the function `main` is slightly modified to accommodate a scenario of two, in this thesis, load cases. The significance of each load case is reflected with the use of a "weighted" gradient, that includes information from the gradient of both of the load cases. The skeleton of the procedure as described in the previous sections is kept, whereas the modifications done in each part of the function `main` are described subsequently.

4.5.1. Pre-processing

In the **User input** part, as defined in subsection §4.2.1, the user has to specify two factors which serve as the weights in the gradient to be used for the design update. These weights, f_1 and f_2, signify the importance of load case 1 and load case 2, respectively, and can take any value between 0 and 1, adding up to 1 in total. Furthermore, the user inputs the boundary conditions for both of the load cases and the initial FE problem is solved for both of them, as well.

4.5.2. Optimization

Inside functions `TO`, `VS` and `VSonly`, the gradients for both of the load cases are calculated; the first one using the FE data for load case 1 and the second one using the FE data for load case 2. However, the design update is based on a "weighted" gradient, expressed as a superposition of the two gradients calculated, making use of the weights f_1 and f_2.

For topology optimization, the design update is modified as

$$\rho_j^{k+1} = \rho_j^k - a_{TO}\left(f_1 G_{j,L_1} + f_2 G_{j,L_2} + \Lambda\right), \tag{4.5.1}$$

whereas for variable stiffness design the design update is

$$V_{1j}^{k+1} = V_{1j}^{k} - a_{VS,1}\left(f_1 G_{1j,L_1} + f_2 G_{1j,L_2}\right), \tag{4.5.2}$$

$$V_{3j}^{k+1} = V_{3j}^{k} - a_{VS,3}\left(f_1 G_{3j,L_1} + f_2 G_{3j,L_2}\right), \tag{4.5.3}$$

where L_1 and L_2 signify load case 1 and load case 2, respectively. Staggered optimization makes use of all the above formulas to update the density and lamination parameter fields, respectively. The FE problem is solved for both load cases, using the updated design.

4.5.3. Post-processing

A modification done in this part involves the compliance at convergence. It is calculated as a "weighted" compliance denoted as "$\text{compliance}_{\text{weighted}}$", i.e.

$$\text{compliance}_{\text{weighted}} = f_1 \text{compliance}_{L_1} + f_2 \text{compliance}_{L_2}, \tag{4.5.4}$$

where compliance_{L_1} and compliance_{L_2} are the compliance values at convergence for load case 1 and load case 2, respectively. Also, the compliance versus number of iterations plot includes the compliance values for both load cases, along with the weighted compliance values.

5

Results, Verification and Discussion

This chapter includes the optimization results for three different designs; a flat composite plate in section §5.1, a flat composite lug in section §5.2 and a flat composite aircraft chair bracket in section §5.3. The results comprise mesh convergence analyses of the first two designs, a parametric analysis of the plate and comparative analyses of the three optimization methods for all of the designs. Finally, a comparison with results from the literature is performed in §5.4, which serves as an additional verification of the methodology used. The material properties used in sections §5.1, §5.2 and §5.3 are listed in Table 5.1.

Material properties	
E_1 [GPa]	134
E_2 [GPa]	7.71
G_{12} [GPa]	4.31
ν_{12}	0.301

Table 5.1: Material properties used in sections §5.1, §5.2 and §5.3.

Regarding the comparative analyses, in order to be able to compare the compliance results obtained from the three optimization methods, the volume of each resulting structure needs to be the same. Since topology and staggered optimization produce a structure with a volume prescribed by the user via the volume fraction, the following relationship holds for the final volumes of the structures from topology (or staggered) optimization V_{TO} and variable stiffness design V_{VS}

$$\begin{aligned} V_{\text{TO}} &= V_{\text{VS}} \\ h_{\text{TO}}\text{volume\_fraction}A_{\text{init}} &= h_{\text{VS}}A_{\text{init}} \\ h_{\text{TO}}\text{volume\_fraction} &= h_{\text{VS}}, \end{aligned} \tag{5.0.1}$$

where h_{TO} and h_{VS} is the laminate thickness used for topology (or staggered) optimization and variable stiffness design, respectively, and A_{init} is the initial area of the structural domain. Therefore, the thickness for variable stiffness design needs to be adjusted, according to the desired volume fraction used for topology and staggered optimization.

5.1. Flat composite plate

The examples in this section were run on a device with specifications listed in Table 5.2.

Device specifications	
Operating system name	CentOS Linux 7 (Core)
System type	64-bit operating system, x64-based processor
Processor	Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz
Threads per core	2
Installed RAM	125 GB

Table 5.2: Device specifications / composite plate models.

5.1.1. Mesh convergence

5.1.1.1. Topology optimization

The case of a flat composite plate with dimensions 50 mm x 20 mm under a point load of 100 N applied on the middle of the right side of the plate was studied initially. The boundary conditions are shown in Figure 5.1. Four different meshes were used for the mesh convergence, with element lengths of 2 mm, 1 mm, 0.5 mm and 0.33 mm. The different meshes used are depicted in Figure 5.2 and the mesh characteristics of each model are listed in Table 5.3.

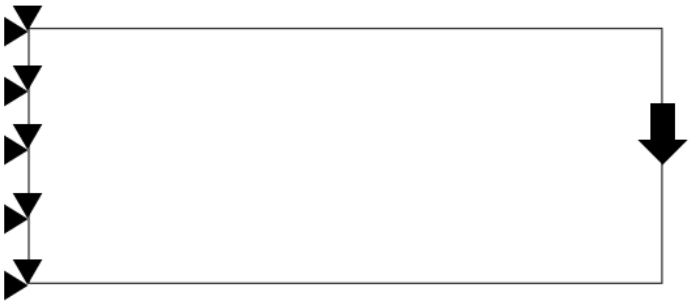
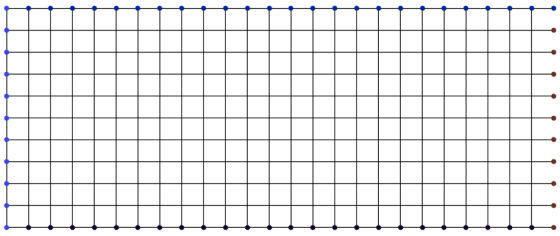
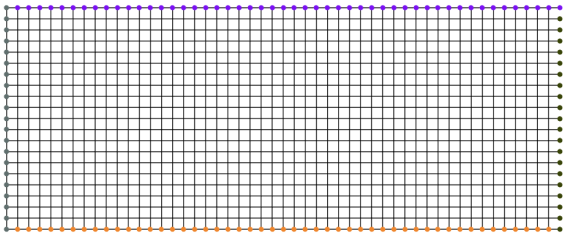


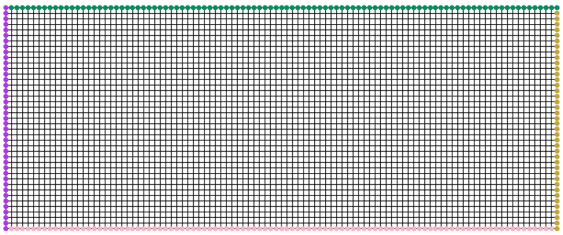
Figure 5.1: Boundary conditions for the flat composite plate.



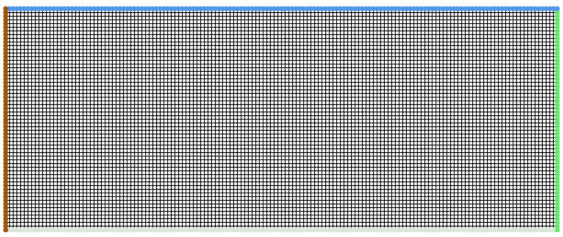
(a) Mesh 1, element length 2 mm.



(b) Mesh 2, element length 1 mm.



(c) Mesh 3, element length 0.5 mm.



(d) Mesh 4, element length 0.33 mm.

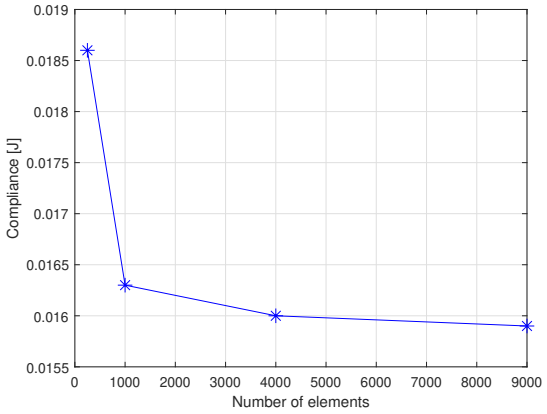
Figure 5.2: Different mesh sizes for the flat composite plate.

	1	**2**	**3**	**4**
Element length [mm]	2	1	0.5	0.33
Number of elements - X direction	25	50	100	150
Number of elements - Y direction	10	20	40	60
Degrees of freedom	572	2142	8282	18422

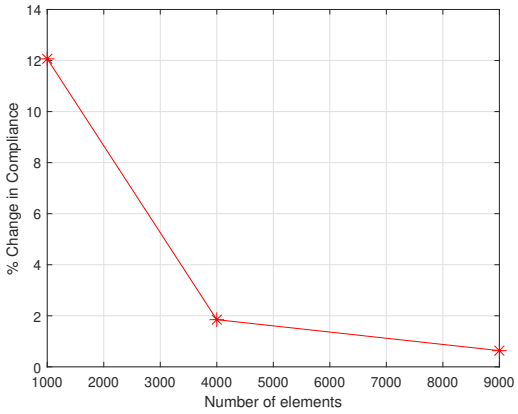
Table 5.3: Mesh characteristics for the different composite plate meshes.

For this study, the maximum penalty power $p_{\max}$ was set to 4, the penalty power step size Δp to 0.1 and the minimum radius for the projection scheme $r_{\min}$ to 2 mm. The desirable volume fraction assigned was 0.5.

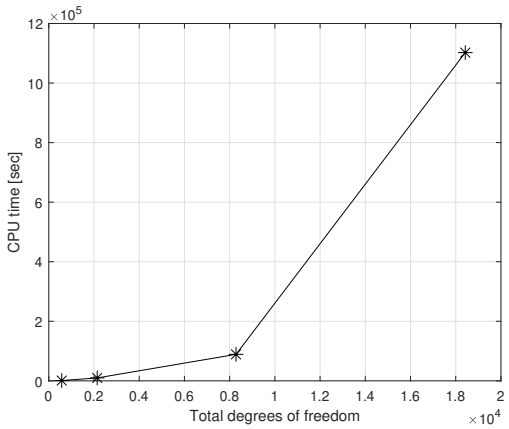
The graph of the compliance at the converged iteration for every model versus the number of elements is shown in Figure 5.3a and the % change in compliance, with respect to the previous mesh, versus the number of elements, in 5.3b. Also, the CPU time at convergence is plotted with respect to the total number of degrees of freedom (Figure 5.3c).



(a) Compliance vs number of elements.



(b) % change in compliance vs number of elements.

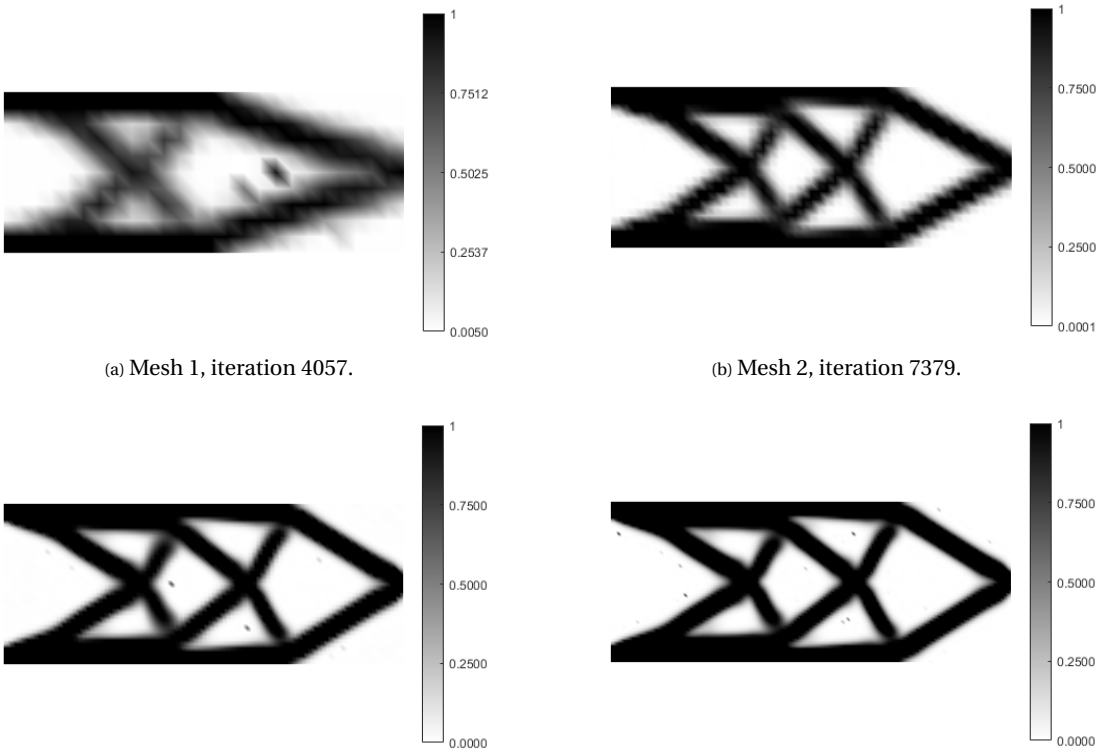


(c) CPU time vs total degrees of freedom.

Figure 5.3: Topology optimization convergence graphs for the flat composite plate.

The results of the converged iteration for every model follow next (Figure 5.4a to 5.4d). The data at convergence are listed in Table 5.4. There is a significant increase in the sharpness of the result as the mesh is refined which, at the same time, means increased computational cost. On the other hand, a more black and white solution favors manufacturing as it requires little to no post-processing. Mesh 3, corresponding to an element size of 0.5 mm, shows good convergence, at a relatively low computational expense and provides a

black and white result. That is the reason why this element size was chosen for the parametric analysis in subsection §5.1.2 and for the comparative analysis in subsection §5.1.3.



(a) Mesh 1, iteration 4057.

(b) Mesh 2, iteration 7379.

(c) Mesh 3, iteration 5109.

(d) Mesh 4, iteration 7144.

Figure 5.4: Plate mesh convergence, topology optimization results.

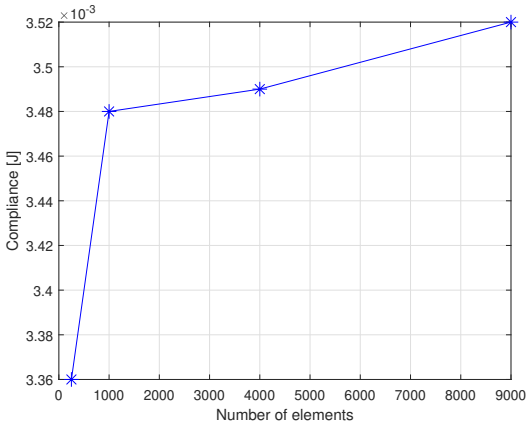
	1	**2**	**3**	**4**
Compliance [J]	0.0186	0.0163	0.0160	0.0159
% change in compliance (w.r.t. previous mesh)	-	12.0662	1.8481	0.6345
Iteration	4057	7379	5109	7144
Tolerance at convergence	1.9698E-08	5.99E-10	2.7397E-09	5.81E-09
CPU time [sec]	988	9809	88994	1102200

Table 5.4: Convergence data for the topology optimization of composite plate models.

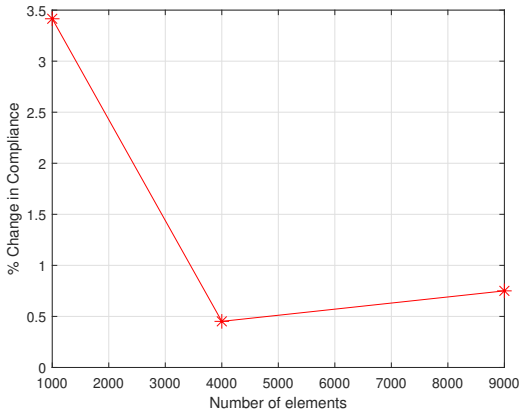
5.1.1.2. Variable stiffness design

For the variable stiffness design mesh convergence analysis of the flat composite plate, the same boundary conditions as in Figure 5.1 and meshes as in Table 5.3 were used.

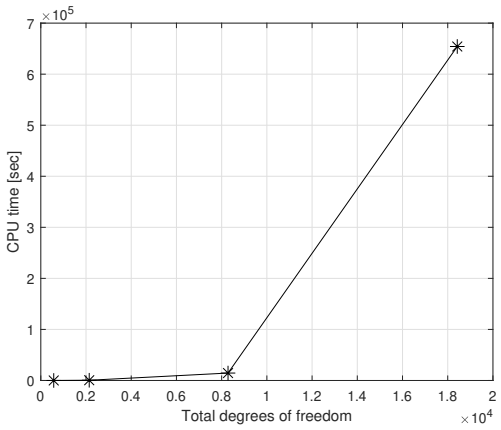
The mesh convergence graphs are shown in Figures 5.5a to 5.5c. The result of the converged iteration for every model is shown in Figure 5.6 and the convergence data are specified in Table 5.5.



(a) Compliance vs number of elements.

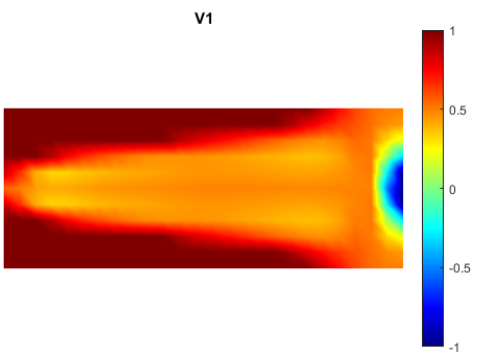


(b) % change in compliance vs number of elements.

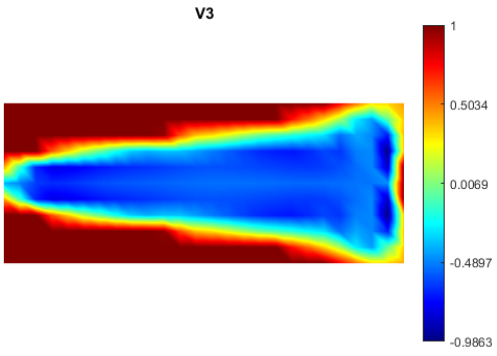


(c) CPU time vs total degrees of freedom.

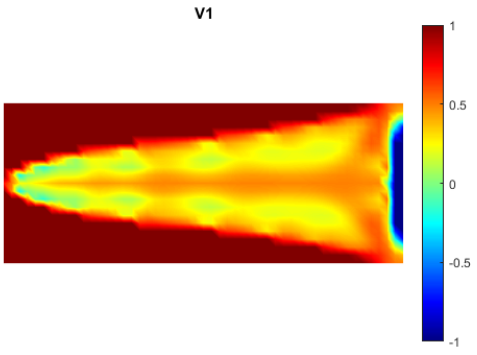
Figure 5.5: Variable stiffness design convergence graphs for the flat composite plate.



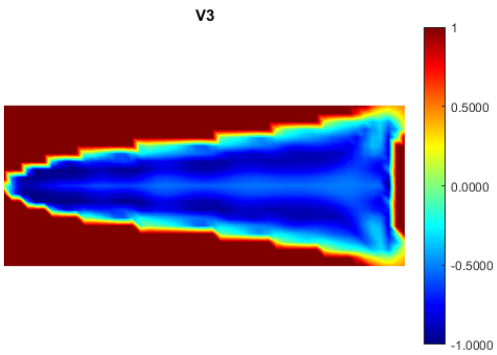
(a) $V_{1\mathbf{A}}$ distribution for mesh 1, iteration 62.



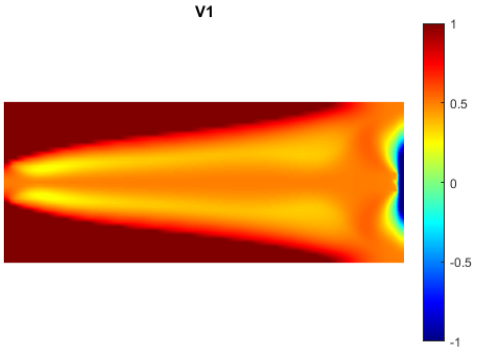
(b) $V_{3\mathbf{A}}$ distribution for mesh 1, iteration 62.



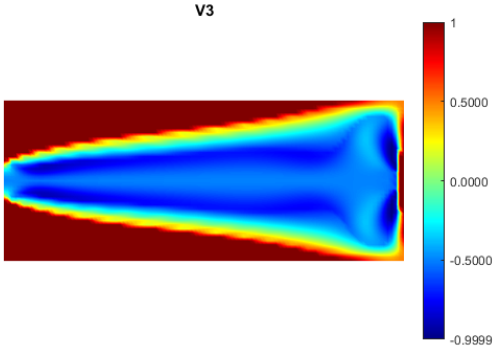
(c) $V_{1\mathbf{A}}$ distribution for mesh 2, iteration 434.



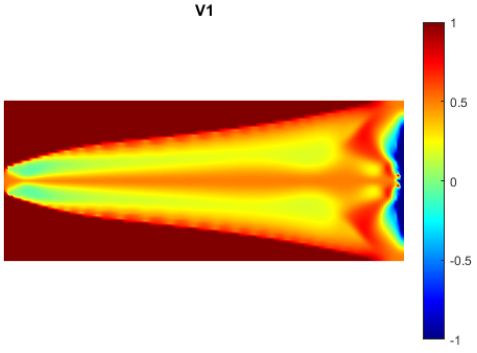
(d) $V_{3\mathbf{A}}$ distribution for mesh 2, iteration 434.



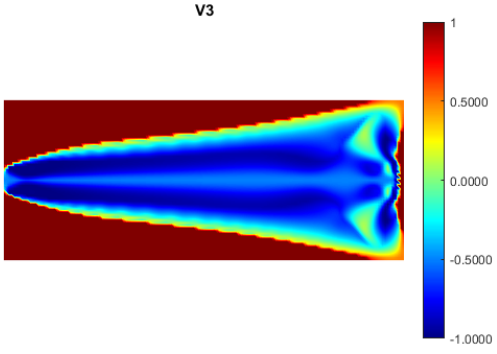
(e) $V_{1\mathbf{A}}$ distribution for mesh 3, iteration 722.



(f) $V_{3\mathbf{A}}$ distribution for mesh 3, iteration 722.



(g) $V_{1\mathbf{A}}$ distribution for mesh 4, iteration 4195.



(h) $V_{3\mathbf{A}}$ distribution for mesh 4, iteration 4195.

Figure 5.6: Plate mesh convergence, variable stiffness design results.

	1	**2**	**3**	**4**
Compliance [J]	0.00336	0.00348	0.00349	0.00352
% change in compliance (w.r.t. previous mesh)	-	3.4152	0.4516	0.7503
Iteration	62	434	722	4195
Tolerance at convergence	1.76E-06	2.94E-08	1.67E-08	9.44E-11
CPU time [sec]	34.4	473.1	14542.0	654210.0

Table 5.5: Convergence data for the variable stiffness design of composite plate models.

In this case, mesh 3 shows good convergence of the compliance value and provides a fine representation of the solution. However, judging from Figure 5.6, we can conclude that the variable stiffness design solution is mesh dependent, due to the formation of more intricate details with mesh refinement. This is due to the fact that no filtering scheme has been imposed on this optimization method. Nevertheless, mesh 3 can be chosen for the comparative analysis of subsection §5.1.3. This is, firstly, in order to comply with the choice made from the topology optimization mesh convergence results of the plate. This way, the same mesh can be also used for staggered optimization, that involves both topology optimization and variable stiffness design at every iteration. Secondly, mesh 3 is an appropriate choice due to the fact that, with a finer mesh, the presence of complicated details in the solution constitute a challenge for manufacturing.

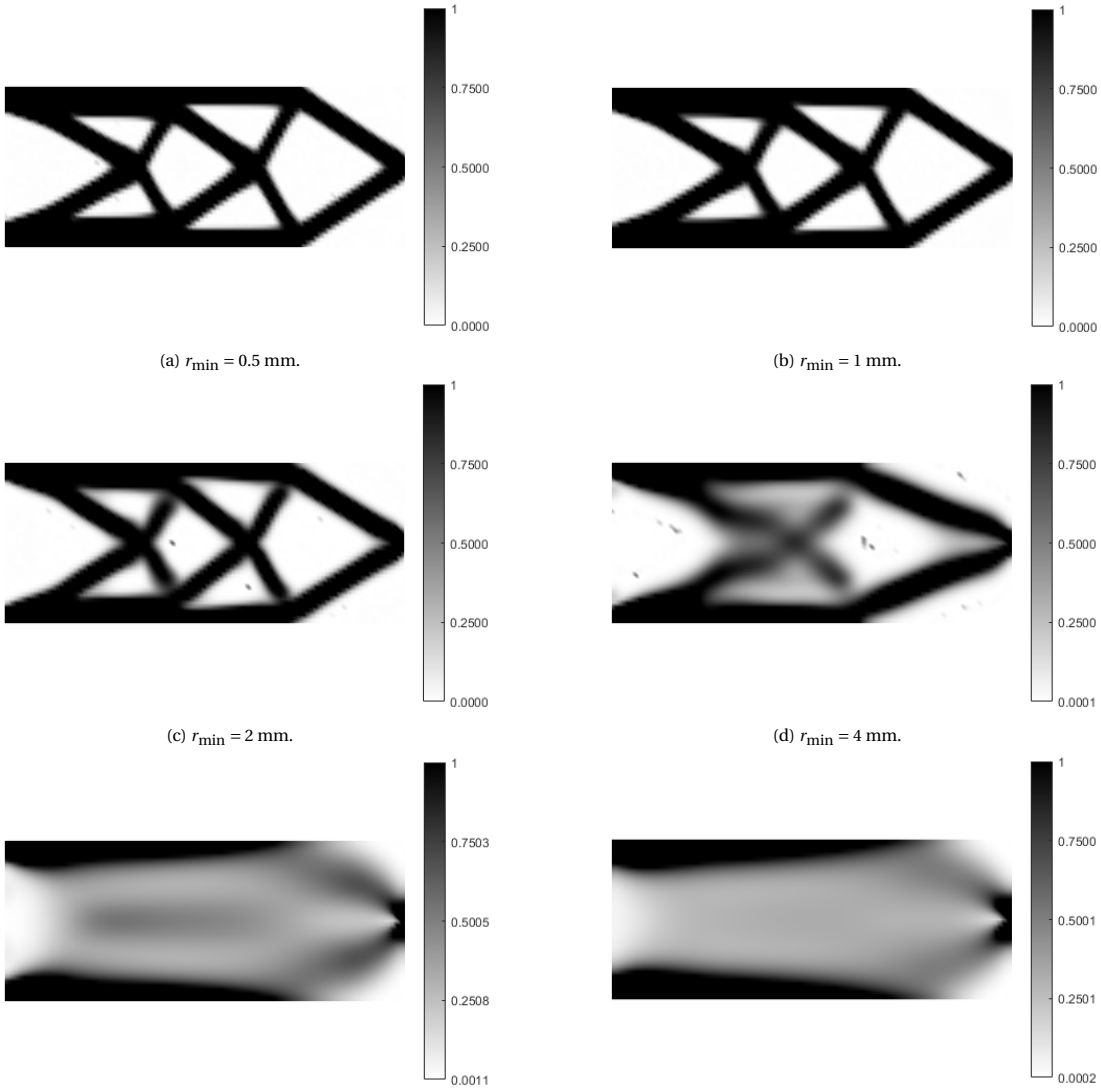
5.1.2. Parametric analysis

In this section the three different parameters used for topology optimization, minimum radius for the projection scheme filter $r_{\min}$, maximum penalty power for the density $p_{\max}$ and continuation method step size Δp, are investigated for their influence on the solution, the compliance, as well as the computational time required to reach the solution. The same boundary conditions as in Figure 5.1 are applied and the assigned volume fraction is 0.5. The graphs of compliance versus the number of topology optimization iterations for each case can be found in Appendix E.

Figure 5.7 depicts the solutions obtained using six different values for $r_{\min}$; 0.5 mm, 1 mm, 2 mm, 4 mm, 8 mm and 12 mm. These results were obtained keeping the other two parameters fixed, i.e. $p_{\max} = 4$ and $\Delta p = 0.1$. Table 5.6 quantifies the results obtained with the different $r_{\min}$ values.

	$r_{\min} = 0.5$	$r_{\min} = 1$	$r_{\min} = 2$	$r_{\min} = 4$	$r_{\min} = 8$	$r_{\min} = 12$
Compliance [J]	0.0116	0.0125	0.0160	0.0264	0.0532	0.0719
% change in compliance (w.r.t. previous)	-	7.3799	28.2629	64.6778	101.9064	35.1125
Iteration	9620	7864	5109	7397	1327	5992
Tolerance at convergence	1.15E-10	1.12E-09	2.74E-09	1.08E-08	5.59E-09	3.26E-09
CPU time [s]	177140	141800	88994	-	-	-
% change in CPU time (w.r.t. previous)	-	-19.9503	-37.2398	-	-	-

Table 5.6: Results data using different $r_{\min}$ values.



(a) $r_{min} = 0.5$ mm.

(b) $r_{min} = 1$ mm.

(c) $r_{min} = 2$ mm.

(d) $r_{min} = 4$ mm.

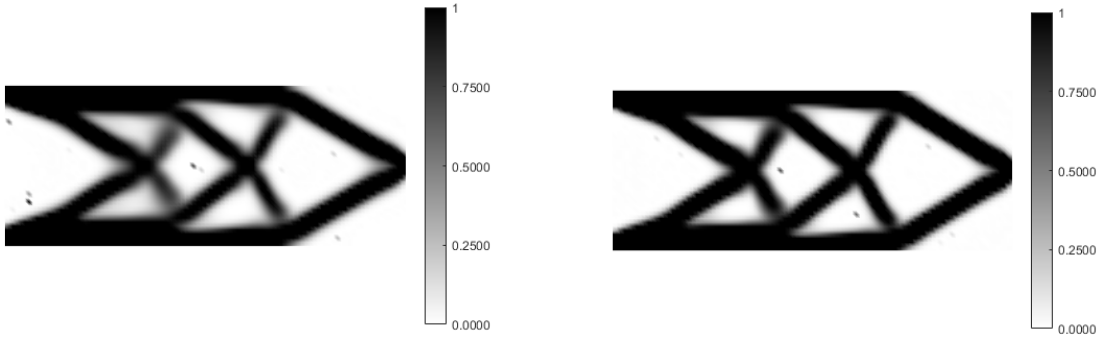
(e) $r_{min} = 8$ mm.

(f) $r_{min} = 12$ mm.

Figure 5.7: Solutions obtained using different r_{min} values.

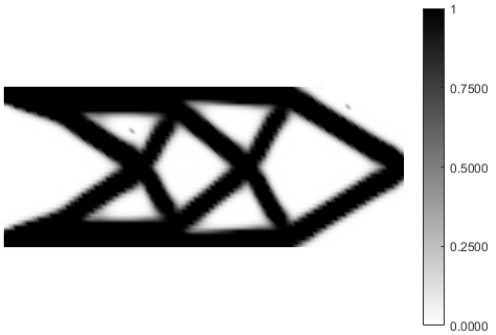
What is initially observed from Figure 5.7 is the fact that increasing r_{min} prevents the structure from creating distinct members in the middle. This is due to the fact that, for the computation of an element density value, more nodal values are taken into account in this way. Therefore, the information obtained for the element is not local anymore and the wide range of nodal values considered results in a gray representation, rather than a black or white one. This is also intertwined with the fact that the compliance increases as r_{min} increases; the information at each node is taken into account for the computation of more element density values. Moreover, by decreasing r_{min} the solution tends to a black and white one and reduces irregularities, i.e. the black spots observed in Figures 5.7c and 5.7d. Regarding the manufacturable solutions, i.e. the ones depicted in Figures 5.7a to 5.7c, the computational time required to achieve the minimum tolerance for a certain number of iterations for the cases with more distinct members, i.e. with $r_{min} = 0.5$ mm and $r_{min} = 1$ mm, is higher than the one with $r_{min} = 2$ mm.

Figure 5.8 depicts the solutions obtained using three different values for p_{max}; 3, 4 and 5. These results were obtained keeping the other two parameters fixed, i.e. $r_{min} = 2$ and $\Delta p = 0.1$.



(a) $p_{max} = 3$

(b) $p_{max} = 4$



(c) $p_{max} = 5$

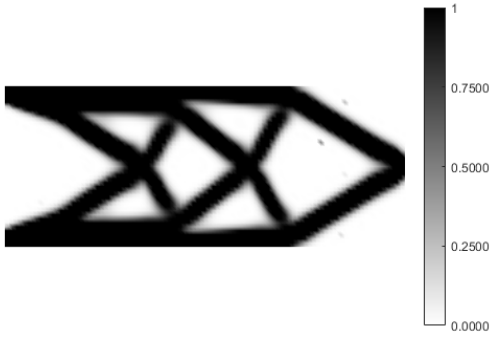
Figure 5.8: Solutions obtained using different p_{max} values.

	$p_{max} = 3$	$p_{max} = 4$	$p_{max} = 5$
Compliance [J]	0.0144	0.0160	0.0174
% change in compliance (w.r.t. previous)	-	11.02282	8.51454
Iteration	13599	5109	13448
Tolerance at convergence	4.15E-09	2.7397E-09	1.81E-09
CPU time [s]	281290	88994	269820
% change in CPU time (w.r.t. previous)	-	-68.3622	203.1890

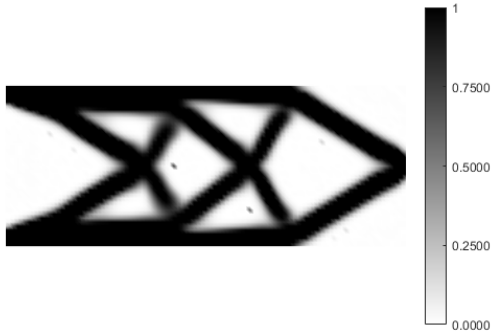
Table 5.7: Results data using different p_{max} values.

Table 5.7 quantifies the results obtained with different values for p_{max}. Looking at Figure 5.8, increasing the penalty power p_{max} provides a more distinct structure for the solution, and thus, easier to manufacture. p_{max} also has an effect on irregularities, which seem to reduce by increasing its value. As expected, the compliance value increases with the increase of the power, since the strain energy is directly related to the SIMP stiffness tensor that increases according to the density penalization. In this case, the solution using $p_{max} = 4$ leads to a minimum tolerance faster than the other two, for a certain number of iterations run. This is subject to the specific case solved each time but, in general, using a lower p_{max} value while keeping the rest of the parameters constant, the convergence is slower. This can be observed at the corresponding graphs in Appendix E.

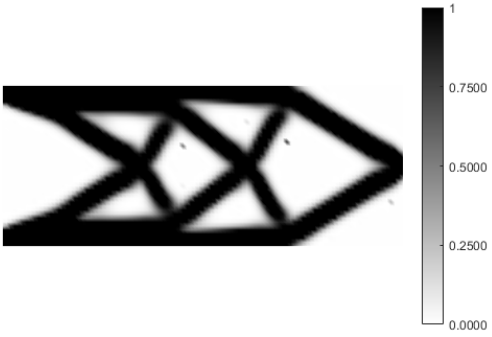
Last, the continuation method step size was investigated. Figure 5.9 shows the solutions obtained using three different values for Δp; 0.05, 0.1 and 0.5. These results were obtained keeping the other two parameters fixed, i.e. $r_{min} = 2$ and $p_{max} = 4$.



(a) $\Delta p = 0.05$



(b) $\Delta p = 0.1$



(c) $\Delta p = 0.5$

Figure 5.9: Solutions obtained using different Δp values.

	$\Delta p = 0.05$	$\Delta p = 0.1$	$\Delta p = 0.5$
Compliance [J]	0.01591	0.01601	0.01592
% change in compliance (w.r.t. previous)	-	0.6430	-0.5658
Iteration	11919	5109	8399
Tolerance at convergence	2.18E-10	2.74E-09	3.64E-09
CPU time [s]	238940	88994	167880
% change in CPU time (w.r.t. previous)	-	-62.7547	88.6419

Table 5.8: Results data using different Δp values.

The data obtained from the results are listed in Table 5.8. The solution obtained does not depend on the continuation step size, as concluded from Figure 5.9. Therefore, the continuation method achieves the purpose of providing a global optimum, irrespective of the step size used. No difference in compliance is observed, except for an expected slight fluctuation in the third and fourth decimals. In this specific case and for a certain number of iterations for all of the examples, the lowest convergence tolerance achieved using $\Delta p = 0.1$ provides the fastest solution between the other two. In general, this does not mean that convergence is achieved faster with this value. Looking at the corresponding graphs in Appendix E, all three solutions start converging after approximately the same number of iterations.

5.1.3. Comparative analysis

5.1.3.1. Single load case

For the case of the flat composite plate, mesh 3 was chosen for the comparative analysis study, as previously mentioned. The boundary conditions for the single load case are depicted in Figure 5.1 and the parameters used in each optimization method are listed in Table 5.9.

	Topology optimization	**Variable stiffness design**	**Staggered optimization**
Δp	0.1	-	0.1
$p_{\max}$	4	-	4
$r_{\min}$ [mm]	1	-	1
Volume fraction	0.5	-	0.5
h [mm]	1	0.5	1
PowerAddition_TO_step	-1	-	-1
PowerAddition_TO_lamda	-3	-	-3
PowerAddition_VS_step	-	-1	-1

Table 5.9: Optimization parameters for the comparative analysis of the flat composite plate for a single load case.

The converged results for all of the optimization methods are depicted in the following figures and quantified in Table 5.10.

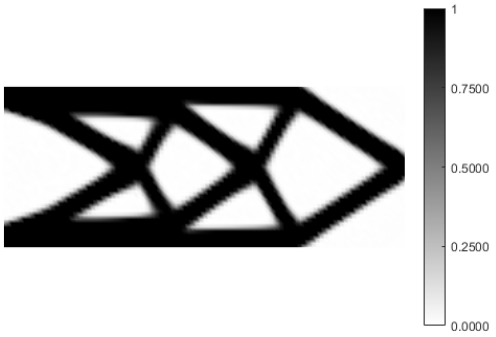
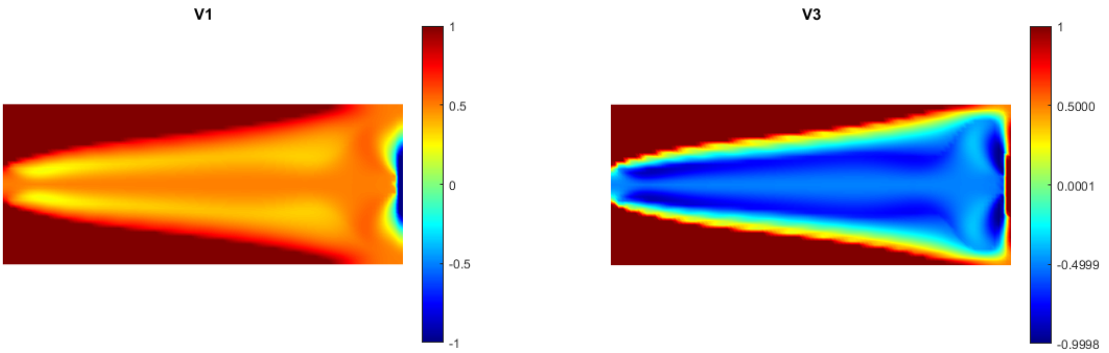


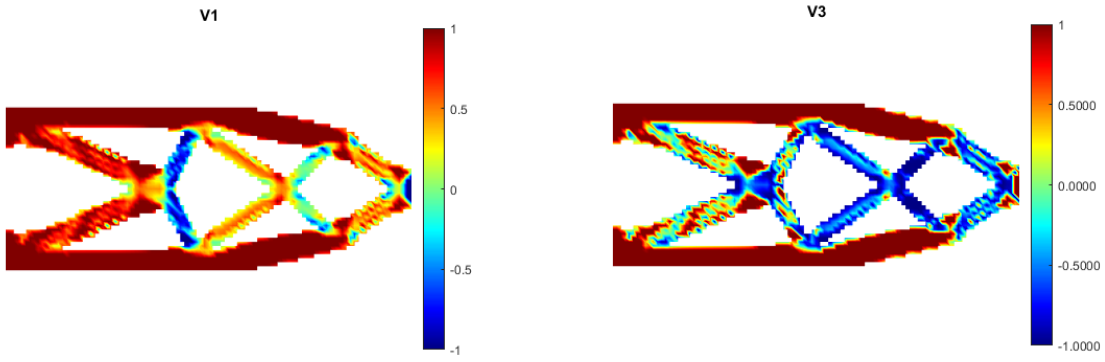
Figure 5.10: Plate comparative analysis, single load case: topology optimization.



(a) $V_{1\mathbf{A}}$ distribution.

(b) $V_{3\mathbf{A}}$ distribution.

Figure 5.11: Plate comparative analysis, single load case: variable stiffness design.



(a) $V_{1\mathbf{A}}$ distribution.

(b) $V_{3\mathbf{A}}$ distribution.

Figure 5.12: Plate comparative analysis, single load case: staggered optimization.

	Topology optimization	**Variable stiffness design**	**Staggered optimization**
Compliance [J]	0.0125	0.0070	0.0082
% change in compliance (w.r.t. topology optimization)	-	-44.0581	-34.6830
Iteration	1849	722	9260
Tolerance at convergence	1.46E-09	1.67E-08	4.64E-08
CPU time [sec]	141800	24596	311790
% change in CPU time (w.r.t. topology optimization)	-	-82.6544	119.8801

Table 5.10: Optimization results for the comparative analysis of the flat composite plate for a single load case.

Both topology and staggered optimization provide a "truss-like" structure, creating beam members in the middle of the domain. It can be observed that topology optimization (Figure 5.10) results in a different geometry than the one obtained with the staggered optimization (Figure 5.12), with the latter one having more "curve-shaped" features. Variable stiffness design places more 0° fibers at the top and bottom part of the structure, whereas, at the loading location, it reinforces the part with more 90° fibers, following the loading direction. The middle area is filled mostly with fibers of approximately 22.5° and fibers close to that value, which implies that the load carrying significance of this area is low. It can also be observed that staggered optimization places the fibers mostly along the beam members. This property accommodates the fact that curve-shaped beams are created on the right side of the structure with staggered optimization; the part is stiffened in those areas by allowing the fibers to follow the direction of the beams. This advantageous feature is not possible with topology optimization. However, the mesh refinement numerical issues of varible stiffness design are evident in this case.

According to Table 5.10, variable stiffness design provides a 44.0581% reduction in compliance with respect to topology optimization, whereas that percentage is lower for staggered optimization, 34.683%. Thus, a stiffer and, at the same time, lighter structure can be achieved for the composite plate subjected to this specific point load, optimizing only the lamination parameters. Variable stiffness design converges much faster than topology optimization (82.6544%), whereas a significant increase in CPU time is noticed with the staggered optimization, where two optimization processes take place and two FE problems are solved in the same iteration. The percentage increase in CPU time is 119.8801%.

5.1.3.2. Double load case

A double load case was also investigated for the flat composite plate. Load case 1 is the same as the one previously solved, depicted in Figure 5.13 for consistency, under a point load of 100 N, and load case 2 is depicted in Figure 5.14 under two point loads of equal magnitudes of 100 N, as well. The two load cases were chosen to be of the same significance and were assigned equal weights, i.e. $f_1 = f_2 = 0.5$. Mesh 3 was used for this example, as well, and the parameters for each optimization method are listed in Table 5.11.

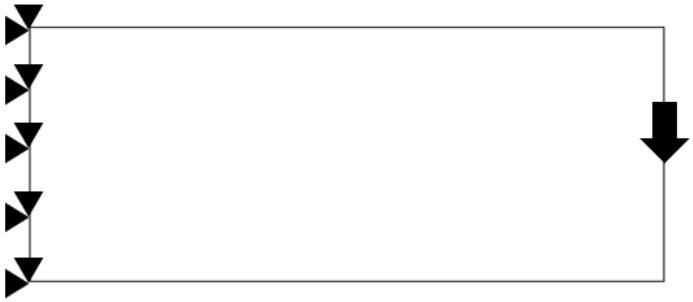


Figure 5.13: Boundary conditions for the flat composite plate: load case 1.

Figure 5.14: Boundary conditions for the flat composite plate: load case 2.

	Topology optimization	**Variable stiffness design**	**Staggered optimization**
Δp	0.1	-	0.1
$p_{\max}$	4	-	4
$r_{\min}$ [mm]	1	-	1
Volume fraction	0.5	-	0.5
h [mm]	1	0.5	1
PowerAddition_TO_step	-1	-	-1
PowerAddition_TO_lamda	-3	-	-3
PowerAddition_VS_step	-	-1	-1

Table 5.11: Optimization parameters for the comparative analysis of the flat composite plate for a double load case.

The optimized results follow in Figures 5.15 to 5.17 and the data at convergence are listed in Table 5.12.

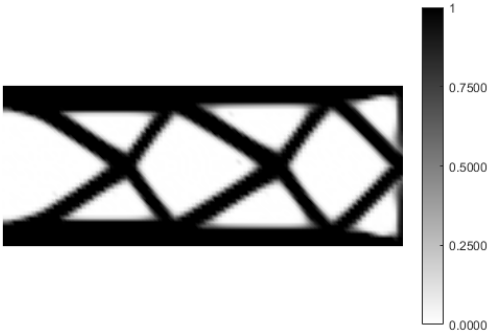
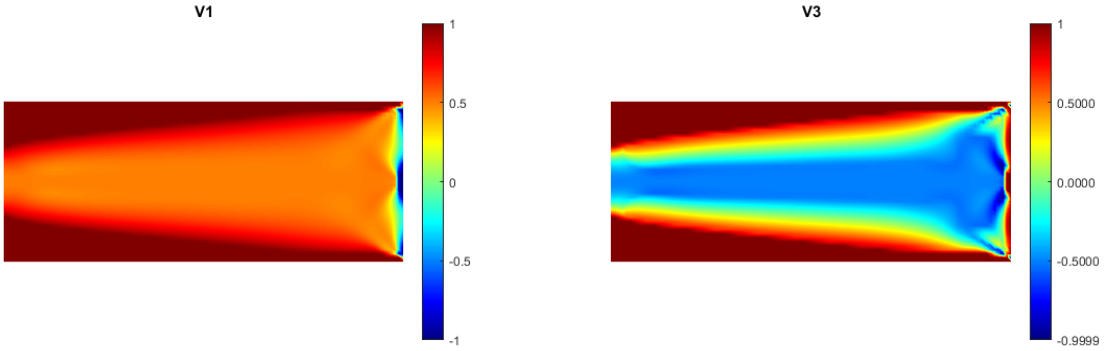
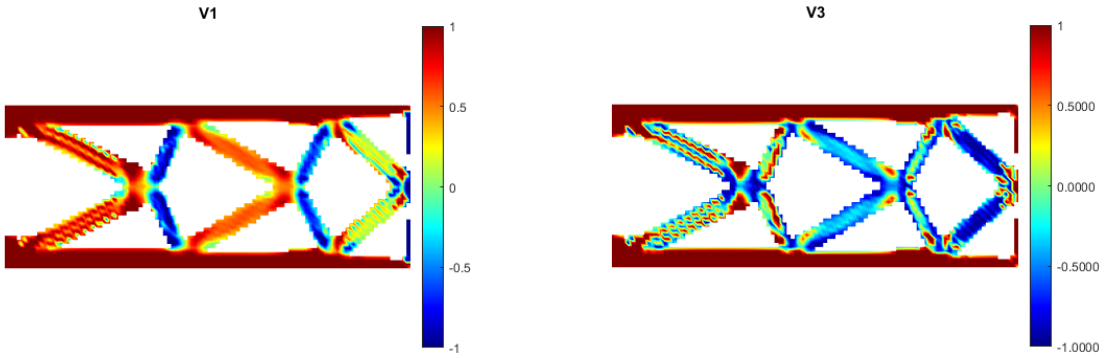


Figure 5.15: Plate comparative analysis, double load case: topology optimization.



(a) $V_{1\mathbf{A}}$ distribution. (b) $V_{3\mathbf{A}}$ distribution.

Figure 5.16: Plate comparative analysis, double load case: variable stiffness design.



(a) $V_{1\mathbf{A}}$ distribution. (b) $V_{3\mathbf{A}}$ distribution.

Figure 5.17: Plate comparative analysis, double load case: staggered optimization

	Topology optimization	**Variable stiffness design**	**Staggered optimization**
Compliance [J]	0.0103	0.0054	0.0060
% change in compliance (w.r.t. topology optimization)	-	-47.5728	-41.7476
Iteration	12183	8463	11833
Tolerance at convergence	1.95E-09	1.26E-10	3.29E-09
CPU time [sec]	416679	339968	693954
% change in CPU time (w.r.t. topology optimization)	-	-18.4100	66.5440

Table 5.12: Optimization results for the comparative analysis of the flat composite plate for a double load case.

In this case, the geometries created by topology and staggered optimization differ from the classic one obtained from the single load case. Additional members are generated to carry the second load case, both at the top and bottom and at the right side of the structure. Moreover, the two designs obtained from topology and staggered optimization are quite similar in terms of geometry. Variable stiffness design extends the area on the right side of the plate where more 90° fibers are placed to accommodate the additional loading, whereas staggered optimization places again the fibers primarily along the beams of the structure. With staggered optimization, on the right side of the structure, the additional transverse beams are assigned a higher percentage of 90° fibers to accommodate load case 1, whereas the additional horizontal beams are assigned a higher percentage of 0° fibers to accommodate load case 2.

According to Table 5.12, variable stiffness design provides a more efficient structure than topology optimization by 47.5728%, reducing the CPU time by 18.41%. On the other hand, staggered optimization comes second, reducing the compliance by 41.7476% but increasing the computational time by 66.5440%. In this scenario, the difference in compliance between variable stiffness design and staggered optimization, 11.1111%, is smaller than the difference noted between them in the single load case, 17.1429%. The graphs of the compliance versus the number of iterations for the composite plate for both the single and the double load case can be found in Appendix E.

5.2. Flat composite lug

The examples in this section were run on a device with specifications listed in Table 5.13.

Device specifications	
Operating system name	Microsoft Windows 10 Enterprise
System type	64-bit operating system, x64-based processor
Processor	Inter(R) Core(TM) i5-4670 CPU @ 3.40 GHz
Threads per core	1
Installed RAM	8 GB

Table 5.13: Device specifications / composite lug models.

5.2.1. Mesh convergence

5.2.1.1. Topology optimization

Next, the case of a flat composite lug design, provided by NLR - Netherlands Aerospace Center, under a constant traction of 100 N/mm was run. The dimensions of the lug can be found in Appendix D and the boundary conditions implemented are illustrated in Figure 5.18. Here, four different meshes were used for the mesh convergence, as well, with average element lengths of 7.8 mm, 4.3 mm, 3 mm and 2.6 mm. The mesh characteristics of each model are listed in Table 5.14 and the different meshes used are depicted in Figure 5.19.

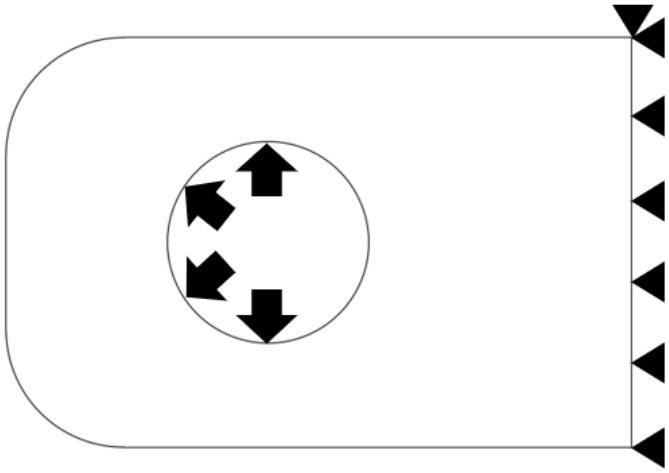
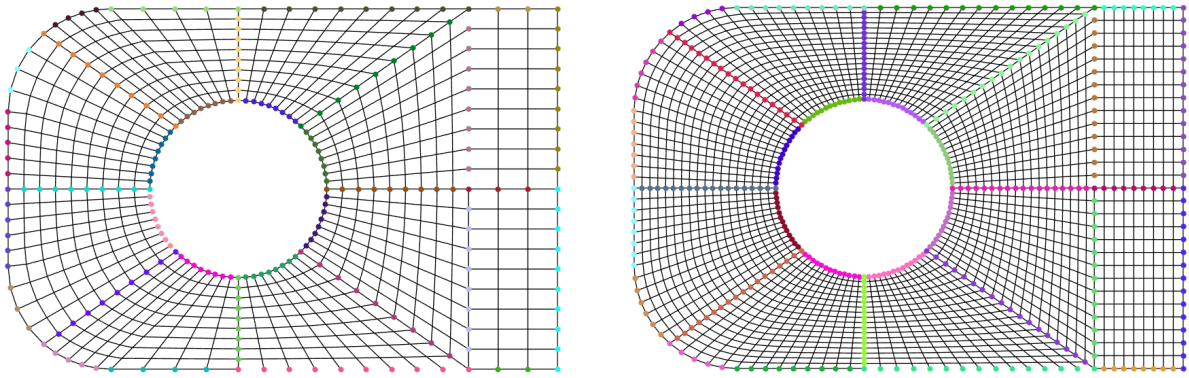


Figure 5.18: Boundary conditions for the flat composite lug.

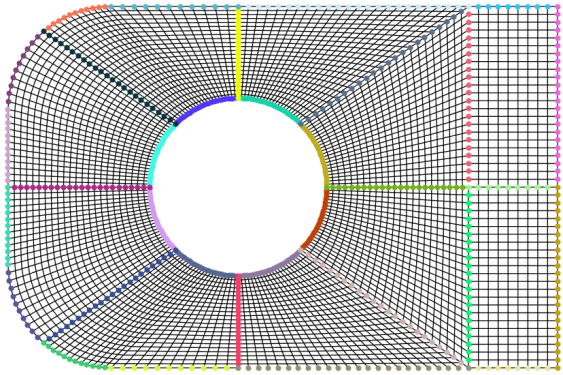
	1	**2**	**3**	**4**
Average element length [mm]	7.8	4.3	3	2.6
Total number of elements	702	2268	4646	6318
Degrees of freedom	1556	4780	9680	13088

Table 5.14: Mesh characteristics for the different composite lug meshes.

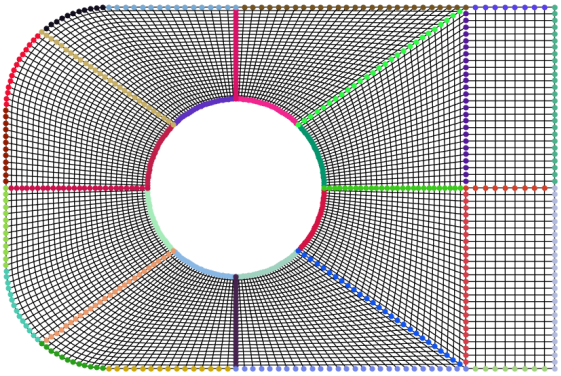


(a) Mesh 1, average element length 7.8 mm.

(b) Mesh 2, average element length 4.3 mm.



(c) Mesh 3, average element length 3 mm.

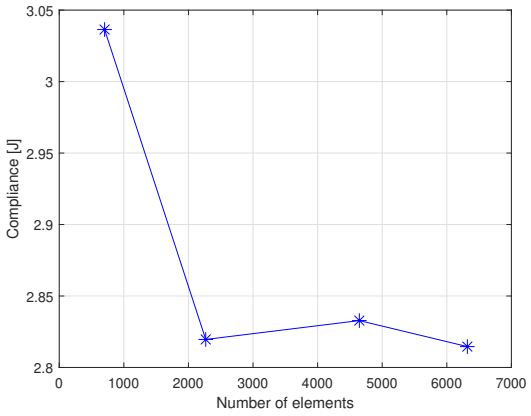


(d) Mesh 4, average element length 2.6 mm.

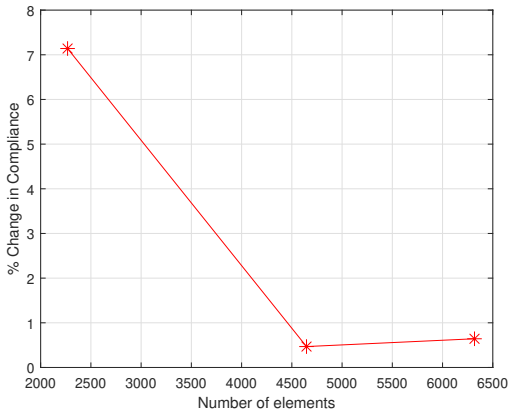
Figure 5.19: Different mesh sizes for the flat composite lug.

The maximum penalty power $p_{\max}$ was set to 4, the penalty power step size Δp to 0.1 and the minimum radius for the projection scheme $r_{\min}$ to 2 mm. The desirable volume fraction assigned was 0.5.

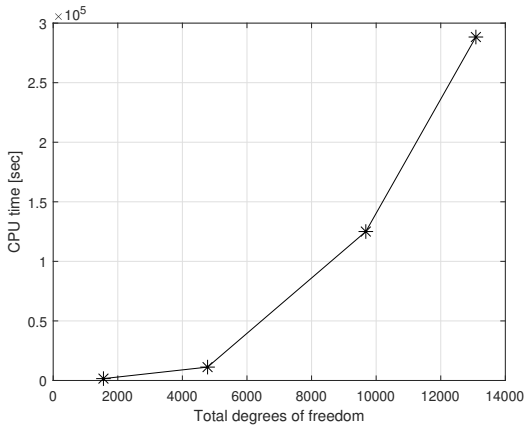
The curves of the compliance at the converged iteration for every model versus the number of elements, the % change in compliance with respect to the number of elements and the CPU time versus the total number of degrees of freedom are depicted in Figure 5.20.



(a) Compliance vs number of elements.



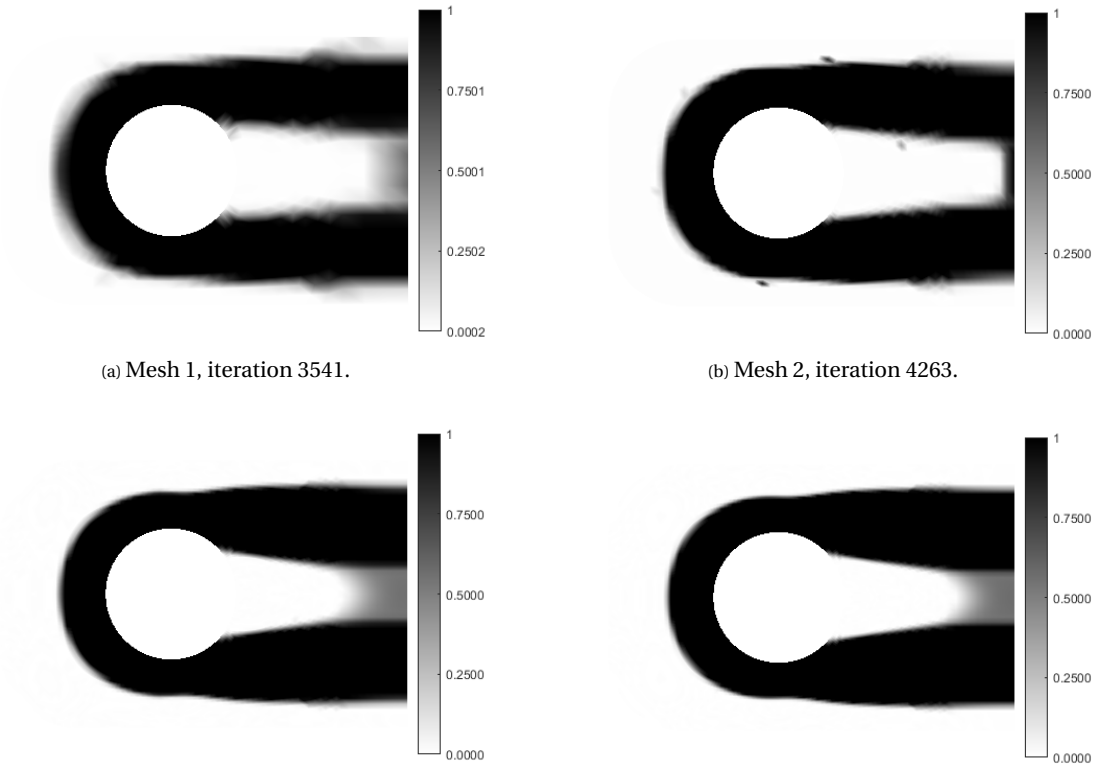
(b) % change in compliance vs number of elements.



(c) CPU time vs total degrees of freedom.

Figure 5.20: Topology optimization convergence graphs for the flat composite lug.

The results of the converged iteration for every model follow next (Figure 5.21a to 5.21d). The data at convergence are listed in Table 5.15. Looking at the graph of Figure 5.20a, we can conclude that the design has already converged using mesh 2, thus a slight fluctuation is noticed between meshes 2, 3 and 4 in the second decimal of the compliance value. As expected, the computational time increases significantly with mesh refinement, and, for this case, from mesh 3 and onwards. However, this mesh is chosen for the comparative analysis of section §5.2.2, since it provides a solution with clearer and more detectable boundaries than the rest, as depicted in Figure 5.21c.



(a) Mesh 1, iteration 3541.

(b) Mesh 2, iteration 4263.

(c) Mesh 3, iteration 7757.

(d) Mesh 4, iteration 9145.

Figure 5.21: Lug mesh convergence, topology optimization results.

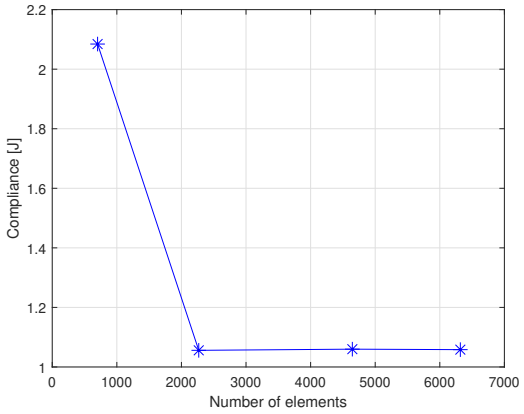
	1	**2**	**3**	**4**
Compliance [J]	3.0364	2.8196	2.8328	2.8146
% change in compliance (w.r.t. previous mesh)	-	7.1400	0.4682	0.6425
Iteration	3541	4263	7757	9145
Tolerance at convergence	8.589E-10	2.3455E-09	3.20E-09	3.36E-09
CPU time [sec]	1570	11233	125030	288330

Table 5.15: Convergence data for the topology optimization of composite lug models.

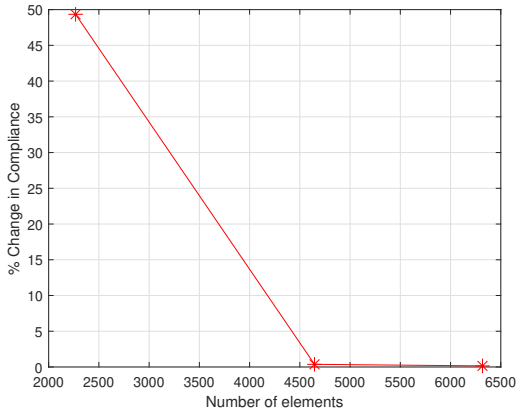
5.2.1.2. Variable stiffness design

For the variable stiffness design mesh convergence analysis of the flat composite lug, the same case as in Figure 5.18 and meshes as in Table 5.14 were used.

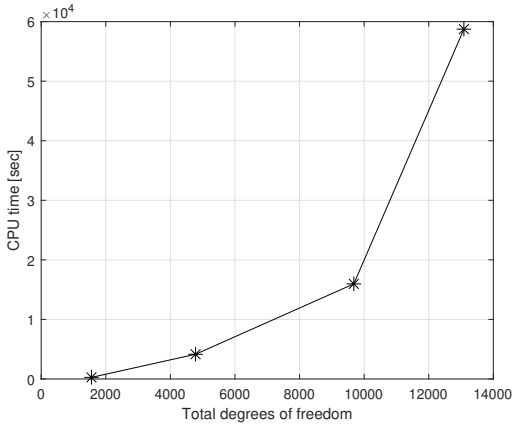
The mesh convergence graphs are shown in Figures 5.22a to 5.22c. The results of the converged iteration for every model are illustrated in Figure 5.23 and the convergence data are specified in Table 5.16.



(a) Compliance vs number of elements.

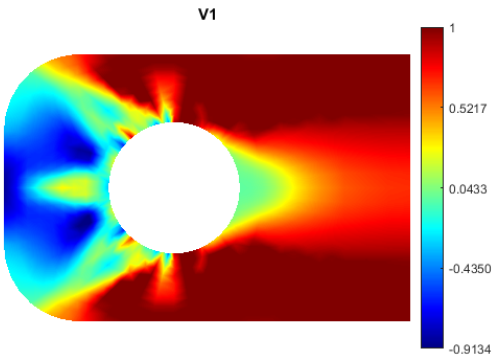


(b) % change in compliance vs number of elements.

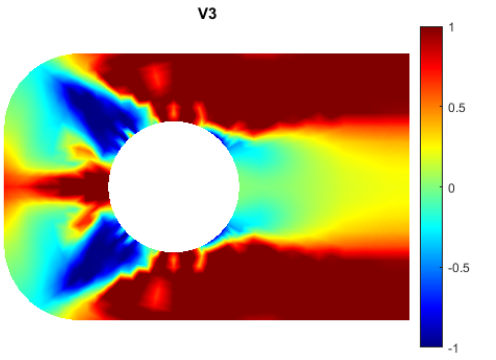


(c) CPU time vs total degrees of freedom.

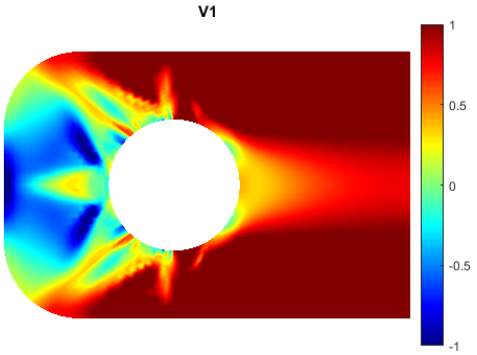
Figure 5.22: Variable stiffness design convergence graphs for the flat composite lug.



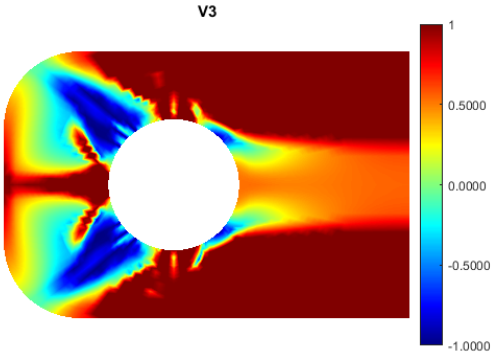
(a) $V_{1\mathbf{A}}$ distribution for mesh 1, iteration 625.



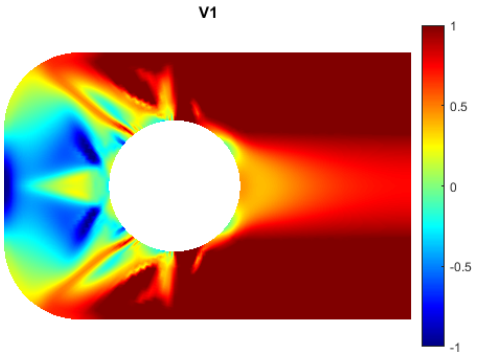
(b) $V_{3\mathbf{A}}$ distribution for mesh 1, iteration 625.



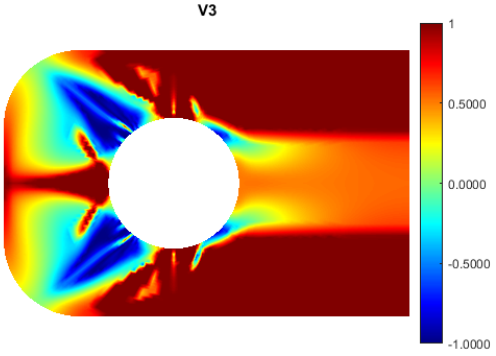
(c) $V_{1\mathbf{A}}$ distribution for mesh 2, iteration 1524.



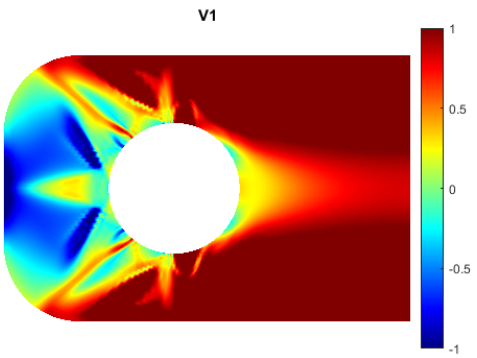
(d) $V_{3\mathbf{A}}$ distribution for mesh 2, iteration 1524.



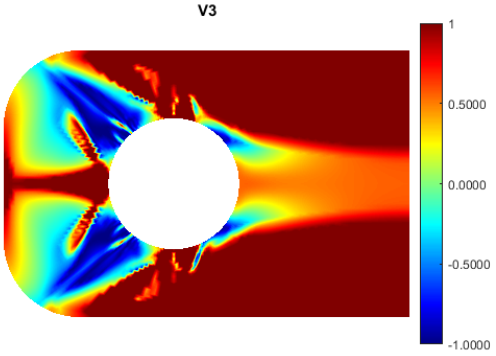
(e) $V_{1\mathbf{A}}$ distribution for mesh 3, iteration 1192.



(f) $V_{3\mathbf{A}}$ distribution for mesh 3, iteration 1192.



(g) $V_{1\mathbf{A}}$ distribution for mesh 4, iteration 2016.



(h) $V_{3\mathbf{A}}$ distribution for mesh 4, iteration 2016.

Figure 5.23: Lug mesh convergence, variable stiffness design results.

	1	**2**	**3**	**4**
Compliance [J]	2.0844	1.0559	1.0598	1.0582
% change in compliance (w.r.t. previous mesh)	-	49.3427	0.3694	0.1510
Iteration	625	1524	1192	2016
Tolerance at convergence	6.99E-08	2.5041E-09	6.73E-10	1.65E-09
CPU time [sec]	271.0	4152.2	15966	58740

Table 5.16: Convergence data for the variable stiffness design of composite lug models.

Judging from the convergence graphs, mesh 3 provides a good compliance convergence and keeps the computational cost as low as possible, while complying with the selection made from the topology optimization mesh convergence results for the lug. That is the reason why mesh 3 was also selected for the staggered optimization of subsection §5.2.2. Here, the mesh dependency is obvious, as well, as more complex details appear when the mesh becomes finer, especially around the area of the circular cutout where the traction is applied.

5.2.2. Comparative analysis

For the case of the flat composite lug, mesh 3 was chosen for the comparative study, as well. The parameters used in each optimization method are listed in Table 5.17.

	Topology optimization	**Variable stiffness design**	**Staggered optimization**
Δp	0.1	-	0.1
$p_{\max}$	4	-	4
$r_{\min}$ [mm]	2	-	2
Volume fraction	0.5	-	0.5
h [mm]	1	0.5	1
PowerAddition_TO_step	-1	-	-1
PowerAddition_TO_lamda	-2	-	-2
PowerAddition_VS_step	-	-1	-1

Table 5.17: Optimization parameters for the comparative analysis of the flat composite lug.

The solutions are illustrated in Figures 5.24 to 5.26 and quantified in Table 5.18.

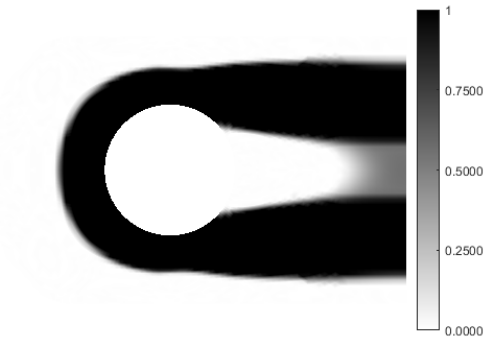
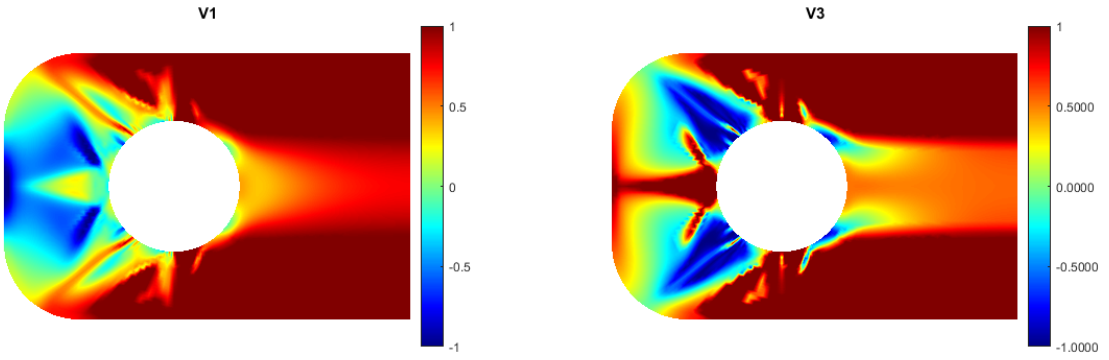
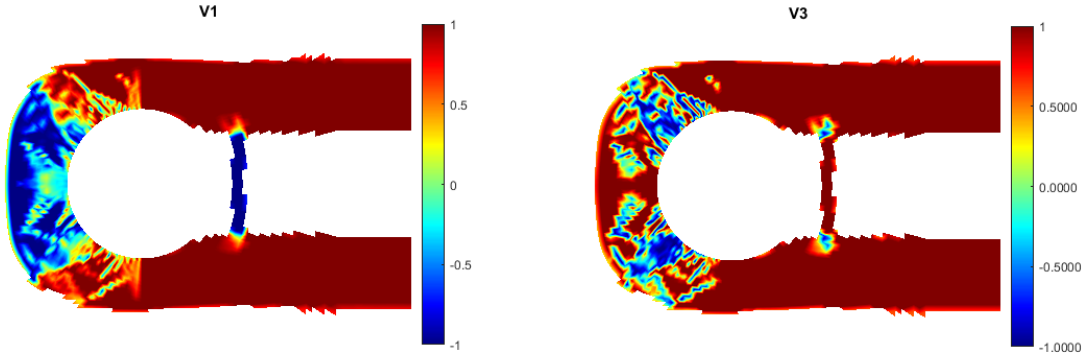


Figure 5.24: Lug comparative analysis: topology optimization.



(a) $V_{1\mathbf{A}}$ distribution. (b) $V_{3\mathbf{A}}$ distribution.

Figure 5.25: Lug comparative analysis: variable stiffness design.



(a) $V_{1\mathbf{A}}$ distribution. (b) $V_{3\mathbf{A}}$ distribution.

Figure 5.26: Lug comparative analysis: staggered optimization.

	Topology optimization	**Variable stiffness design**	**Staggered optimization**
Compliance [J]	2.8328	2.1196	1.642
% change in compliance (w.r.t. topology optimization)	-	-25.1765	-42.0361
Iteration	7757	3375	7619
Tolerance at convergence	3.20E-09	3.65E-11	1.39E-08
CPU time [sec]	125030	47227	196160
% change in CPU time (w.r.t. topology optimization)	-	-62.2275	56.8905

Table 5.18: Optimization results for the comparative analysis of the flat composite lug.

Looking at Figures 5.24 and 5.26, we can see that both methods create a hoop to support the applied traction with two connecting members at the top and bottom of the part. Nevertheless, differences in the obtained geometry are spotted, as well. Topology optimization creates thicker top and bottom members, whereas staggered optimization reduces their thickness while creating a new thin member in the middle, at the same time. This again stems from the fact that staggered optimization provides the flexibility both to subtract material and vary the orientation of the fibers, creating a lighter and at the same time stiff structure. Variable stiffness design places 0° fibers at the top and bottom members, away from the loading, and varies the fiber orientation along the traction location, with 45° fibers being more prominent around the left part of the hole. On the other hand, the fiber orientations resulted from staggered optimization seem to run mostly along the created hoop.

Interestingly, according to Table 5.18, variable stiffness design provides a 25.1765% reduction in compliance with respect to topology optimization, whereas the staggered optimization reduces the compliance by 42.0361%. Of course, the staggered optimization is more costly computationally, compared to the other two optimization methods, as expected.

5.3. Flat composite aircraft chair bracket

The examples in this section were run on a device with specifications listed in Table 5.2.

5.3.1. Comparative analysis: double load case

The last case studied is the design of an aircraft chair bracket, provided by NLR - Netherlands Aerospace Center, with dimensions specified in Appendix D. This part is loaded with a load F equal to 750N, corresponding to the weight of one person, and a moment M of 251Nm. The setup where the bracket is attached is illustrated in Figure 5.27.

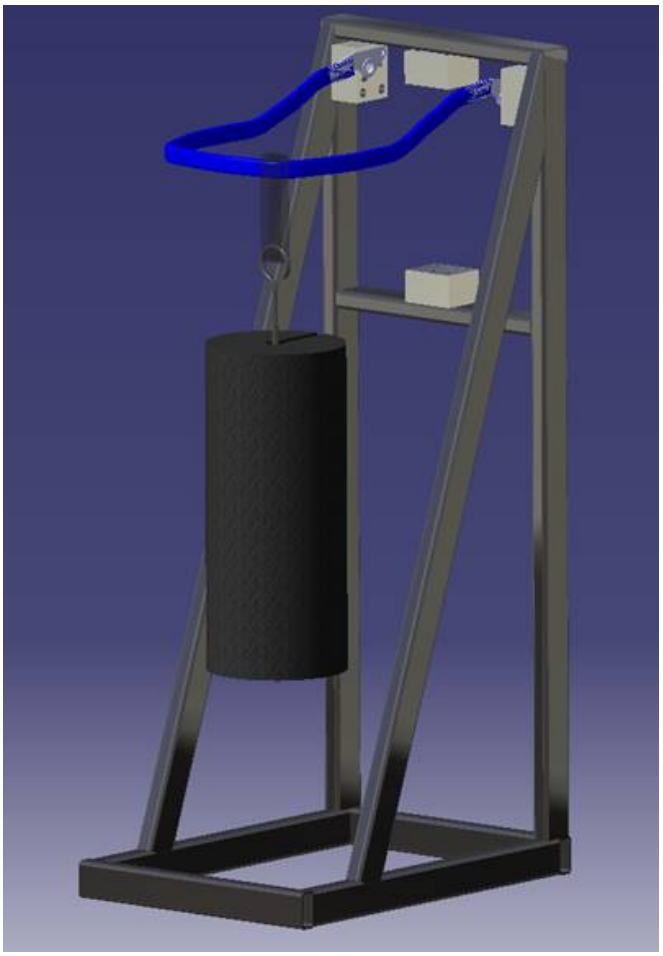


Figure 5.27: Aircraft chair setup.

Load case 1 is the primary load that the bracket has to accommodate. The load of 750N is carried equally by holes 1 and 2 and it is converted to tractions t_1 and t_2, as depicted in Figure 5.28. The moment of 251Nm is converted to two equal forces F_M of the opposite direction which are subsequently translated to tractions t_3 and t_4 on holes 1 and 2. The conversion data and the resulting tractions for load case 1 are listed in Table 5.19. This load case is assigned a weight f_1 equal to 0.6.

Load case 2 is essentially a secondary load, applied to the bracket towards the opposite direction, as depicted in Figure 5.29. The conversion data and the resulting tractions for load case 2 are listed in Table 5.20. This is a rarer scenario and, for that reason, a weight f_2 equal to 0.4 is assigned to this load case.

Three different volume fractions were used for the simulations; 0.3, 0.5 and 0.7. The average element length of the models ran is 1.9 mm, resulting in 2715 elements in total. This is accepted since an average element length of 3 mm was previously deemed sufficient for the case of the lug. The rest of the parameters used for each optimization method are listed in Table 5.21.

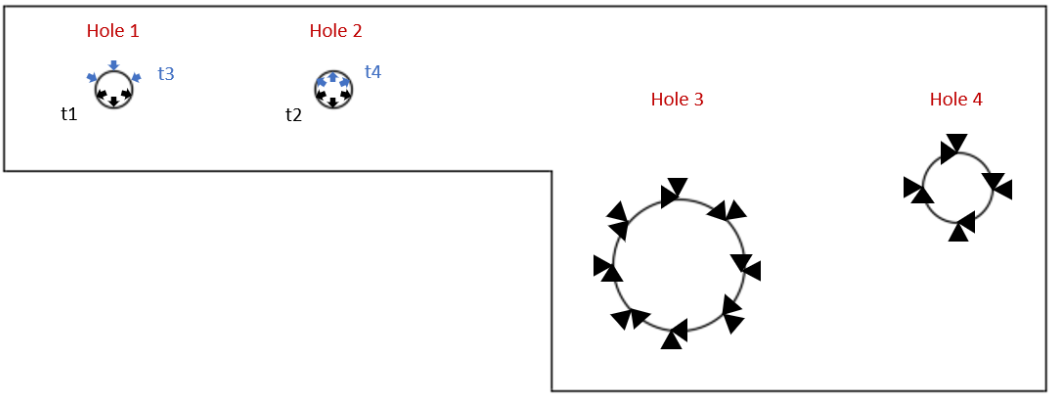


Figure 5.28: Boundary conditions for the flat composite chair bracket: load case 1.

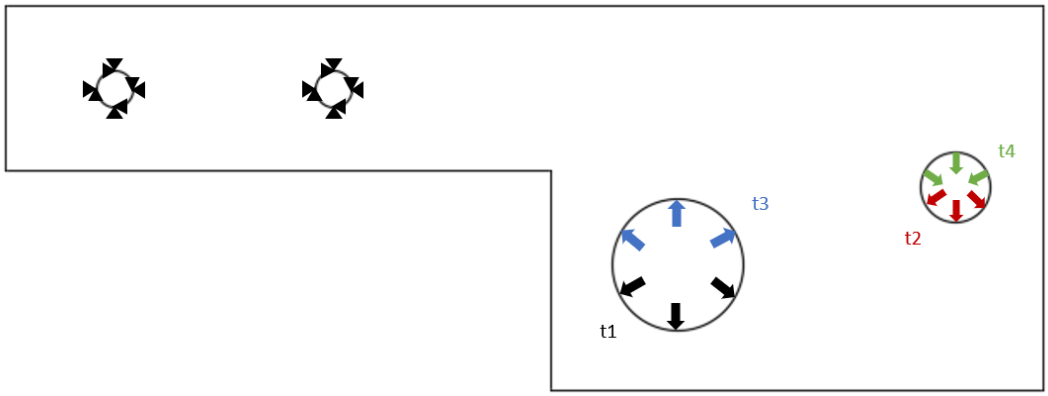


Figure 5.29: Boundary conditions for the flat composite chair bracket: load case 2.

Load case 1		
	Hole 1	Hole 2
M [Nmm]	251000	251000
Distance between holes 1 and 2 [mm]	40	40
F_M [N]	6275	6275
Arc length [mm]	10.3673	10.3673
t* [N/mm] due to *M	**605.3 (t_3)**	**605.3 (t_4)**
F [N]	375	375
t* [N/mm] due to *F	**36.2 (t_1)**	**36.2 (t_2)**

Table 5.19: Load case 1 tractions for the comparative analysis of the chair bracket.

Load case 2		
	Hole 3	Hole 4
M [Nmm]	251000	251000
Distance between holes 3 and 4 [mm]	50.9	50.9
F_M [N]	4931.2377	4931.2377
Arc length [mm]	37.6991	20.4204
t* [N/mm] due to *M	**130.8 (t_3)**	**241.5 (t_4)**
F [N]	375	375
t* [N/mm] due to *F	**9.9 (t_1)**	**18.4 (t_2)**

Table 5.20: Load case 2 tractions for the comparative analysis of the chair bracket.

	Topology optimization	**Variable stiffness design**	**Staggered optimization**
Δp	0.1	-	0.1
$p_{\max}$	4	-	4
$r_{\min}$ [mm]	1	-	1
PowerAddition_TO_step	-1	-	-1
PowerAddition_TO_lamda	-2	-	-2
PowerAddition_VS_step	-	-1	-1

Table 5.21: Optimization parameters for the comparative analysis of the flat composite chair bracket.

The laminate thickness for topology and staggered optimization is kept constant at 1 mm and the thickness for variable stiffness design is adjusted every time, i.e. h = 0.3 mm, 0.5 mm and 0.7 mm for volume fractions of 0.3, 0.5 and 0.7, respectively.

Volume fraction: 0.3

The topology optimization, variable stiffness design and staggered optimization results for a volume fraction of 0.3 are depicted in Figures 5.30, 5.31 and 5.32, respectively. The data at convergence are listed in Table 5.22.

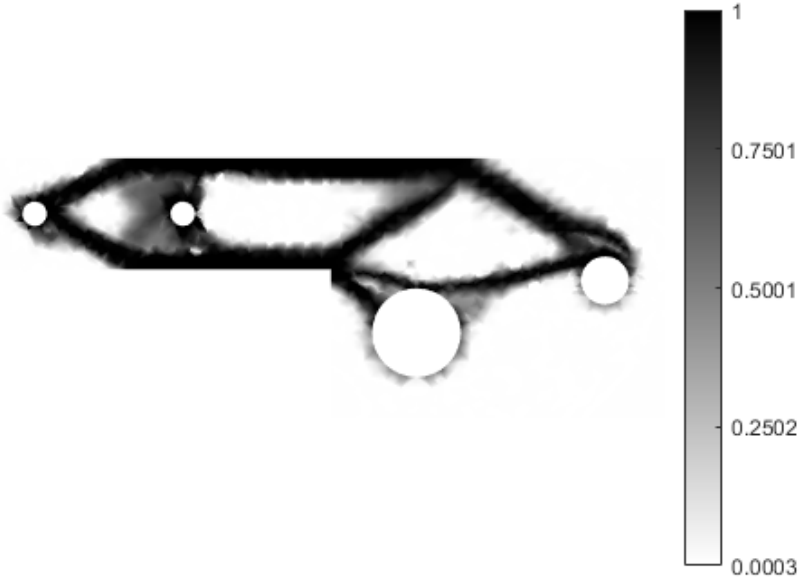
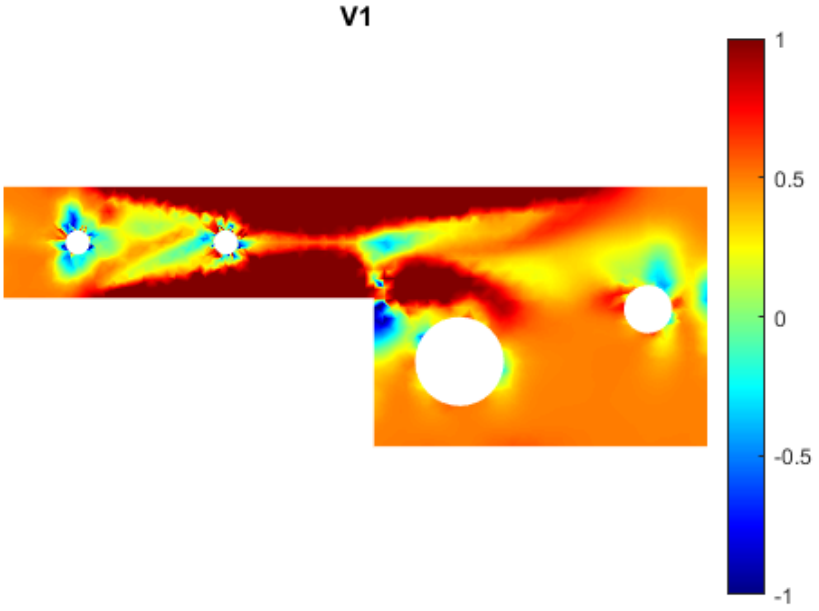
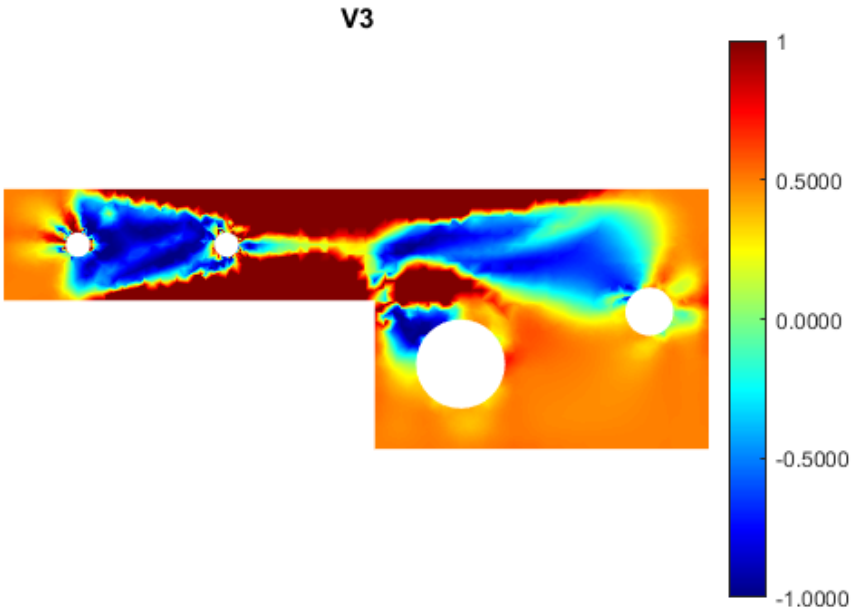


Figure 5.30: Chair bracket comparative analysis: topology optimization, volume fraction = 0.3.

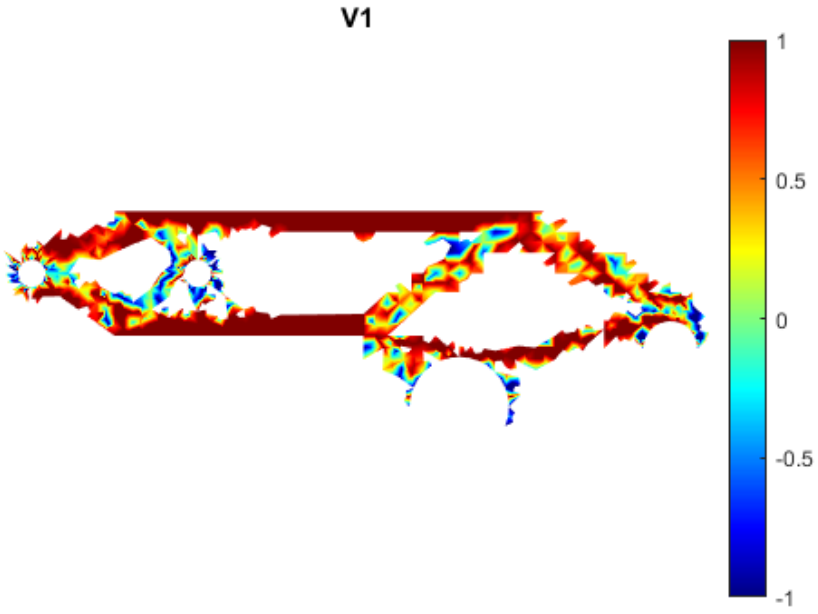


(a) $V_{1\mathbf{A}}$ distribution.

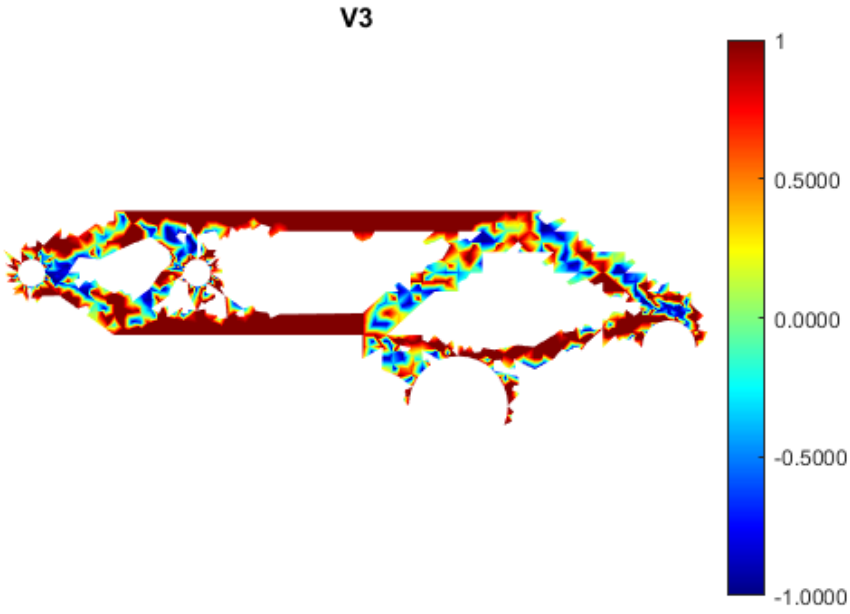


(b) $V_{3\mathbf{A}}$ distribution.

Figure 5.31: Chair bracket comparative analysis: variable stiffness design, $h = 0.3$ mm.



(a) $V_{1\mathbf{A}}$ distribution.



(b) $V_{3\mathbf{A}}$ distribution.

Figure 5.32: Chair bracket comparative analysis: staggered optimization, volume fraction = 0.3.

According to Figure 5.30, topology optimization for a volume fraction of 0.3 creates very thin members in the structure. The areas around holes 1 and 2, where the primary load is applied, are reinforced more than the ones around holes 3 and 4. The thin members are reflected in the staggered optimization (Figure 5.32), as well, where the generated geometry resembles the one created with topology optimization. For variable stiffness design (Figure 5.31), there is an intense variation on the lamination parameters around holes 1 and 2, where the highest loads are applied due to the primary load case. This variation is much more evident in the staggered optimization solution, but that can also be attributed to the mesh dependency of the result at some areas. Moreover, staggered optimization chooses to reinforce holes 1 and 2 more than 3 and 4, due to the higher significance of the primary load case, resulting in absence of material at the bottom areas of holes 3 and 4.

Variable stiffness design achieves a 26.5586% reduction in compliance with respect to topology optimiza-

tion, whereas the staggered optimization produces a much more efficient structure, increasing this percentage to 40.3426%. However, as expected, the computational cost of staggered optimization doubles with respect to topology optimization, whereas variable stiffness design converges faster.

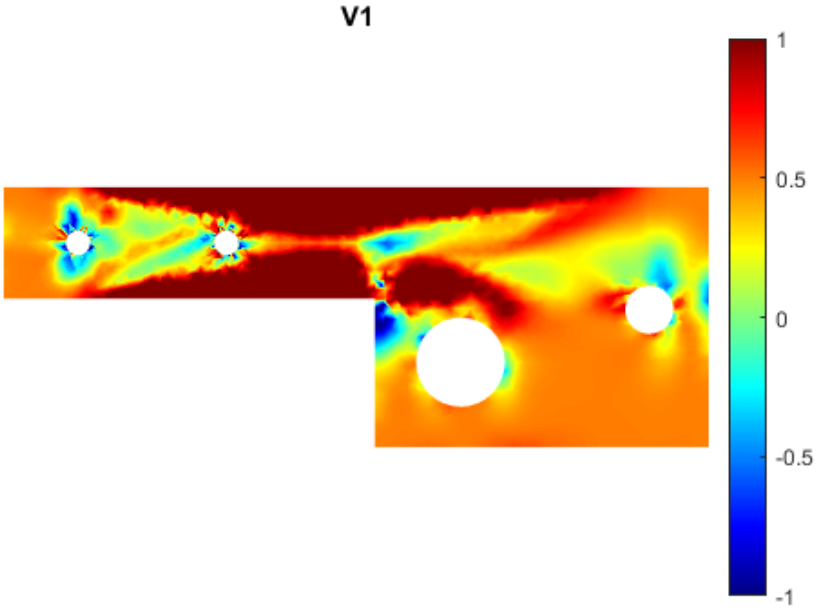
	Topology optimization	**Variable stiffness design**	**Staggered optimization**
Compliance [J]	25.9396	19.0504	15.4749
% change in compliance (w.r.t. topology optimization)	-	-26.5586	-40.3426
Iteration	5374	6977	5574
Tolerance at convergence	5.60E-06	4.82E-08	2.34E-09
CPU time [sec]	123335	109225	186076
% change in CPU time (w.r.t. topology optimization)	-	-11.4407	50.8700

Table 5.22: Optimization results for the comparative analysis of the flat composite chair bracket, volume fraction = 0.3.

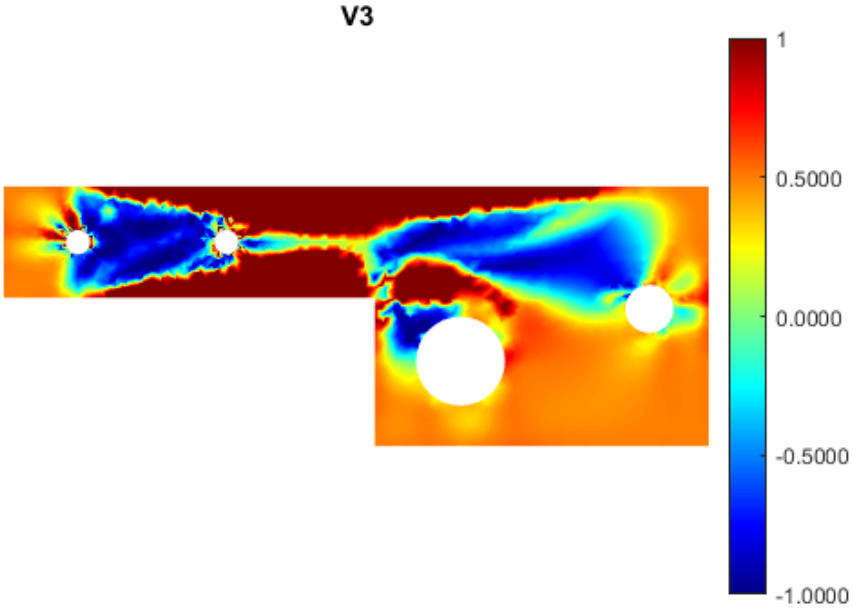
Volume fraction: 0.5

The topology optimization, variable stiffness design and staggered optimization results for a volume fraction of 0.5 are depicted in Figures 5.33, 5.34 and 5.35, respectively. The data at convergence are listed in Table 5.23.

Figure 5.33: Chair bracket comparative analysis: topology optimization, volume fraction = 0.5.

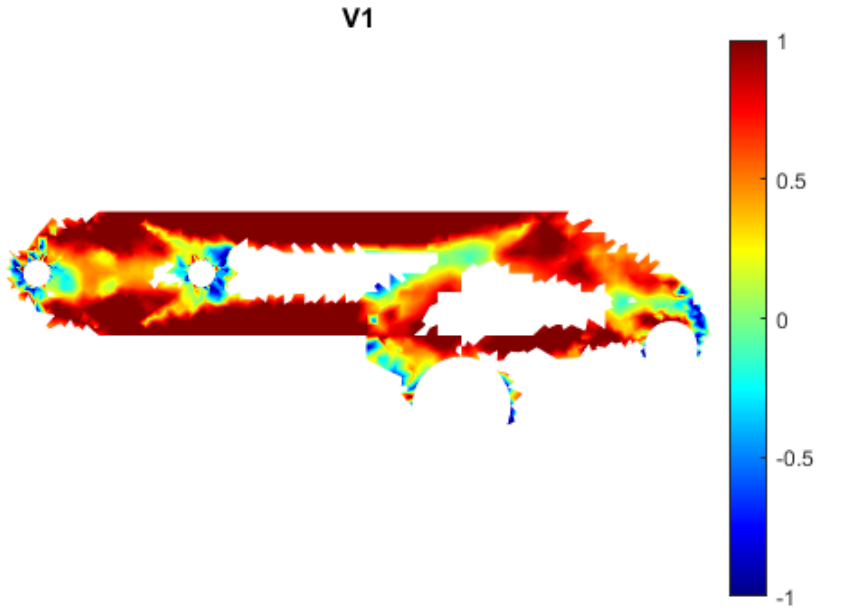


(a) $V_{1\mathbf{A}}$ distribution.

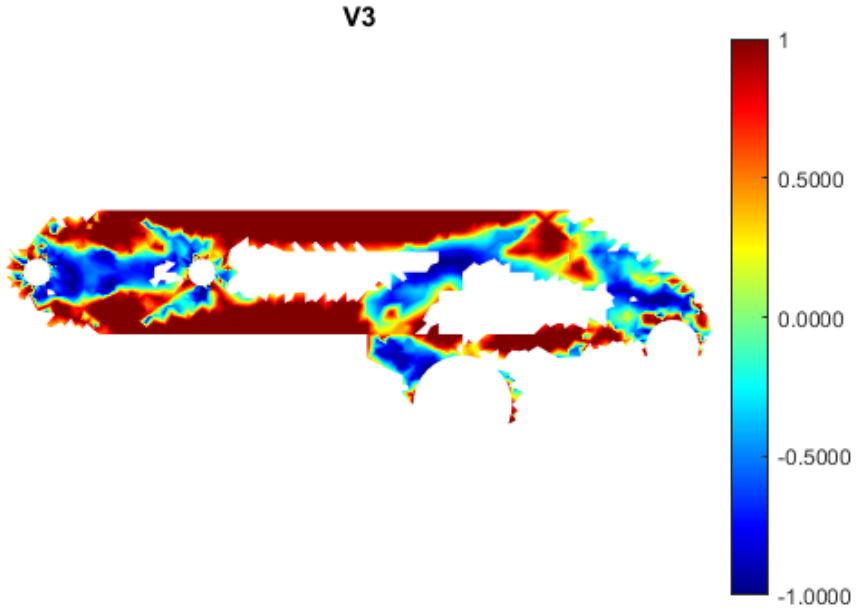


(b) $V_{3\mathbf{A}}$ distribution.

Figure 5.34: Chair bracket comparative analysis: variable stiffness design, $h = 0.5$ mm.



(a) $V_{1\mathbf{A}}$ distribution.



(b) $V_{3\mathbf{A}}$ distribution.

Figure 5.35: Chair bracket comparative analysis: staggered optimization, volume fraction = 0.5.

In this case, due to the increase in the desired volume fraction, thicker members are constructed using topology and staggered optimization, as depicted in Figures 5.33 and 5.35. Here, the reinforcement around holes 1 and 2 is much more evident. Also, the gap between holes 1 and 2 obtained using a lower volume fraction, has almost closed. The other two gaps on the right were assigned thicker and additional members, respectively. The lamination parameter distributions (Figure 5.34) are very similar to the ones obtained with the lower volume fraction.

The reduction of the compliance with variable stiffness design is 13.5409% with respect to topology optimization and, again, the staggered optimization proves to be more efficient than both of the other methods alone, decreasing the compliance by 40.6189%. Another observation here is related to the computational time required for the convergence of the staggered optimization; the minimum tolerance for the specified number of iterations is achieved early enough in the process, which results in a 35.0632% faster convergence

than topology optimization.

	Topology optimization	**Variable stiffness design**	**Staggered optimization**
Compliance [J]	13.2037	11.4158	7.8405
% change in compliance (w.r.t. topology optimization)	-	-13.5409	-40.6189
Iteration	12248	2646	3945
Tolerance at convergence	1.11E-10	5.39E-10	9.04E-10
CPU time [sec]	208847	35633	135618
% change in CPU time (w.r.t. topology optimization)	-	-82.9380	-35.0632

Table 5.23: Optimization results for the comparative analysis of the flat composite chair bracket, volume fraction = 0.5.

Volume fraction: 0.7

The topology optimization, variable stiffness design and staggered optimization results for a volume fraction of 0.7 are depicted in Figures 5.36, 5.37 and 5.38, respectively. The data at convergence are listed in Table 5.24.

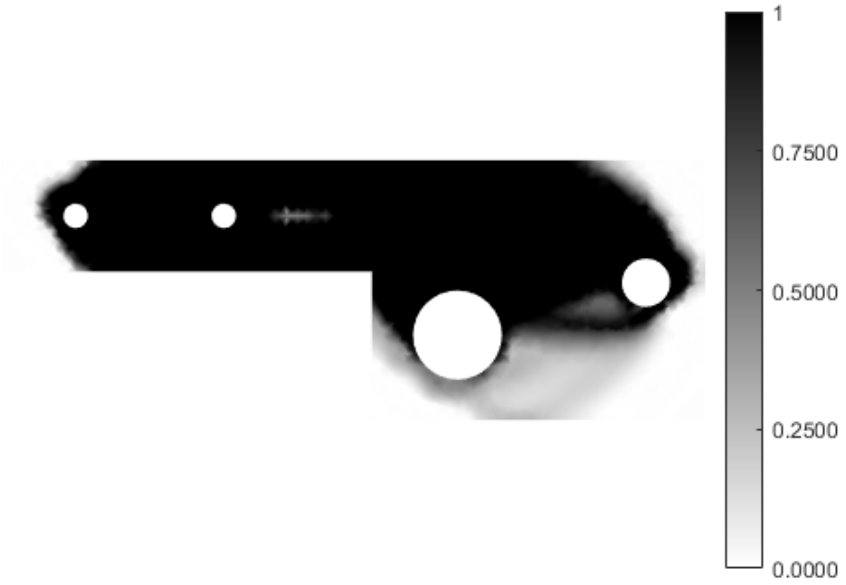
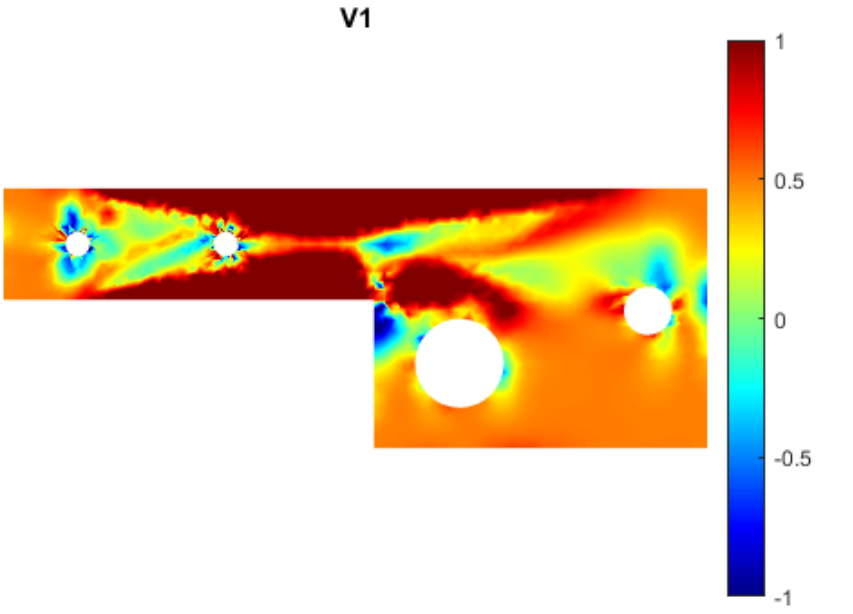
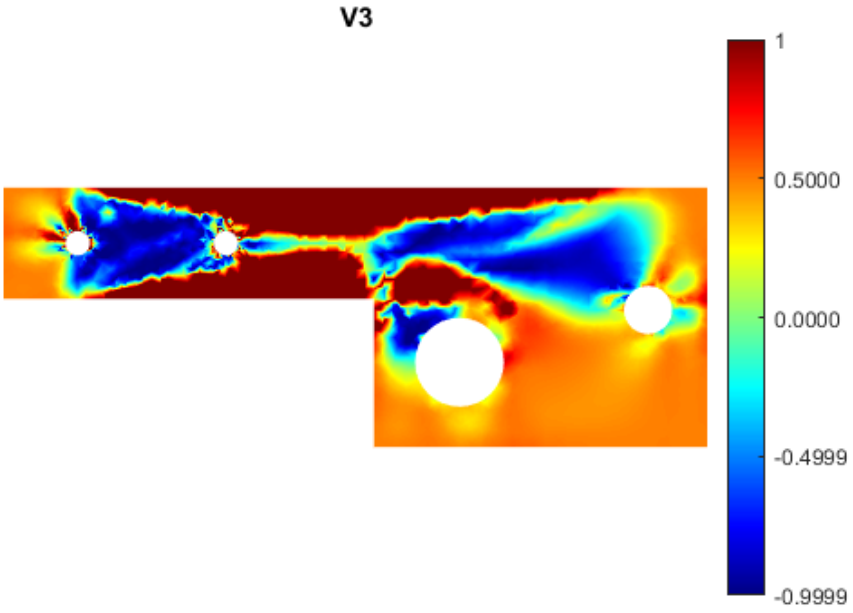


Figure 5.36: Chair bracket comparative analysis: topology optimization, volume fraction = 0.7.

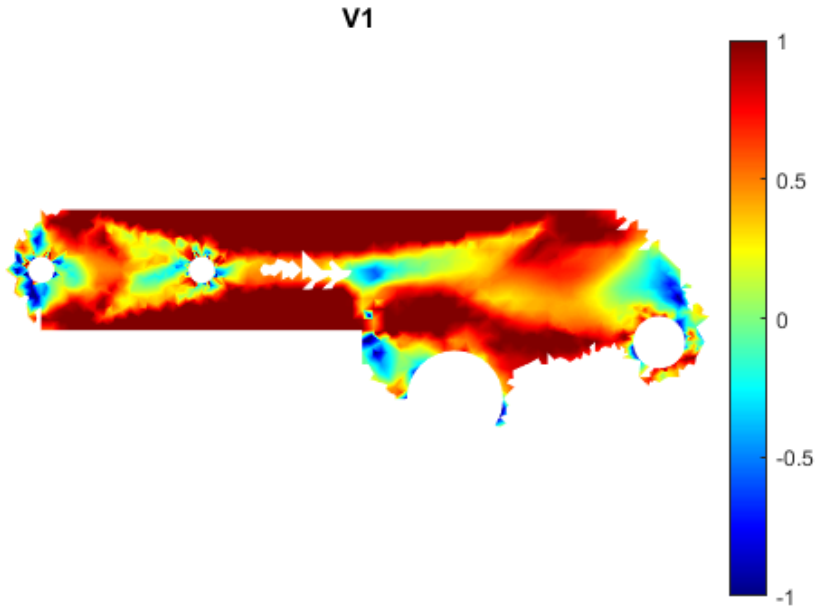


(a) $V_{1\mathbf{A}}$ distribution.

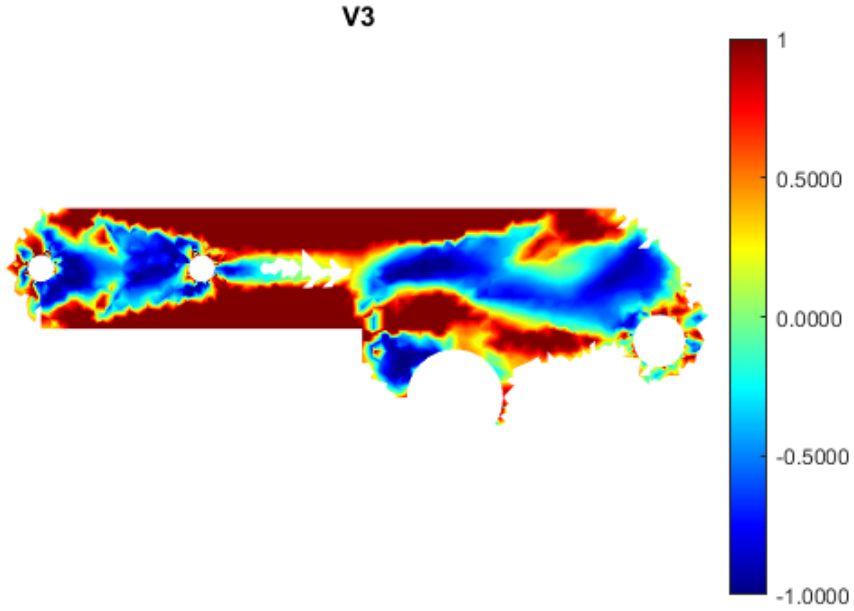


(b) $V_{3\mathbf{A}}$ distribution.

Figure 5.37: Chair bracket comparative analysis: variable stiffness design, $h = 0.7$ mm.



(a) $V_{1\mathbf{A}}$ distribution.



(b) $V_{3\mathbf{A}}$ distribution.

Figure 5.38: Chair bracket comparative analysis: staggered optimization, volume fraction = 0.7.

The topology optimized result for a volume fraction of 0.7, as illustrated in Figure 5.36, requires a certain amount of post-processing due to the present gray areas. This can be attributed to the fact that, for this specific case, the step size is too small to lead to a completely black and white solution in a reasonable time frame. More material is added in this case, compared to the previous volume fraction, that caused two of the openings to close completely, in both topology and staggered optimization (Figure 5.38). Another interesting observation is that, in the result of the staggered optimization, material is placed all around hole 4. Here, hole 4 is chosen to be reinforced, instead of hole 3, since higher loads are applied to it in load case 2, as it can be seen from Table 5.20. As expected, the variation of the fiber angles obtained from variable stiffness design (Figure 5.37) has not changed compared to the rest volume fraction cases.

Variable stiffness design provides an improved solution, with a lower compliance by 28.0336% with respect to topology optimization. The staggered optimization method prevails in this example, as well, as it reduces

the compliance obtained by topology optimization by almost a half and, more particularly, by 47.0557%.

	Topology optimization	**Variable stiffness design**	**Staggered optimization**
Compliance [J]	11.3321	8.1553	5.9997
% change in compliance (w.r.t. topology optimization)	-	-28.0336	-47.0557
Iteration	12793	5954	8907
Tolerance at convergence	1.08E-10	7.89E-10	7.80E-10
CPU time [sec]	198982	92902	258058
% change in CPU time (w.r.t. topology optimization)	-	-53.3112	29.6893

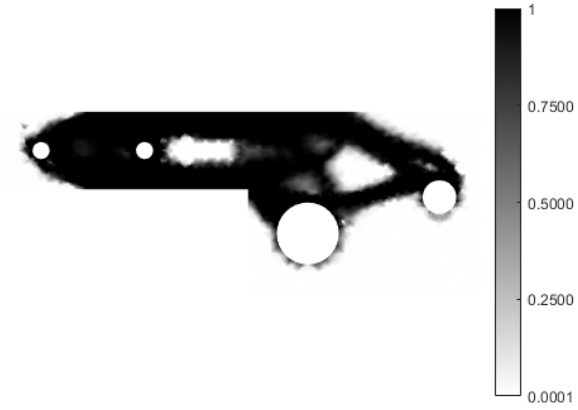
Table 5.24: Optimization results for the comparative analysis of the flat composite chair bracket, volume fraction = 0.7.

5.3.2. Comparative analysis: single vs double load case

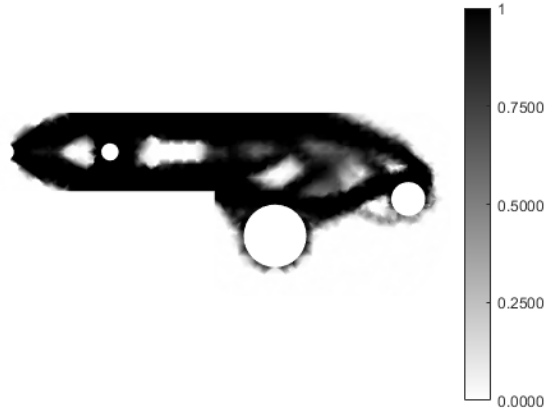
In this subsection, the chair bracket example is demonstrated for three different scenarios; in the first one only load case 1 is applied, the second one is the double load case as solved in **Volume fraction = 0.5** of subsection §5.3.1 and in the third one only load case 2 is applied. The results of the double load case are also presented here for consistency. The volume fraction assigned to topology and staggered optimization is 0.5, the laminate thickness h assigned to variable stiffness design is 0.5 mm and the optimization parameters used are listed in Table 5.21. The purpose of this subsection is to present the evolution of the design when different weights f_1 and f_2 are assigned to the loading cases imposed on a structure. Load case 1 is essentially a scenario where the weight of the primary load, f_1, is equal to 1 and the weight of the secondary load, f_2, is equal to 0. This way, we have a single load case. On the contrary, in load case 2, f_1 is assigned the value 0 and f_2 the value 1, resulting in a single load case, as well.

The topology optimization, variable stiffness design and staggered optimization results for each scenario are depicted in Figures 5.39, 5.40 and 5.41, respectively.

(a) Load case 1.

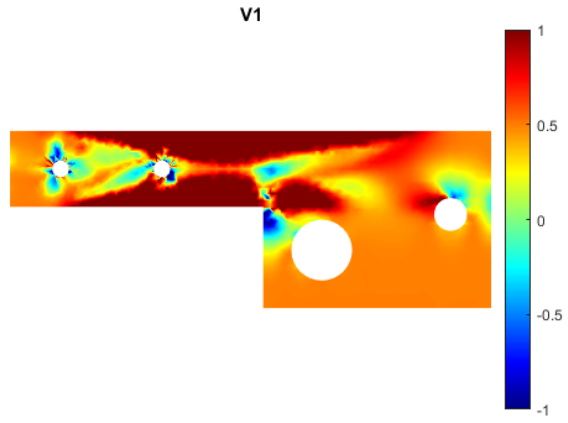


(b) Double load case.

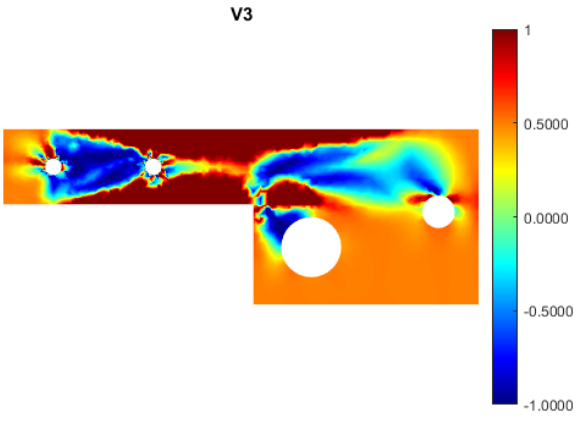


(c) Load case 2.

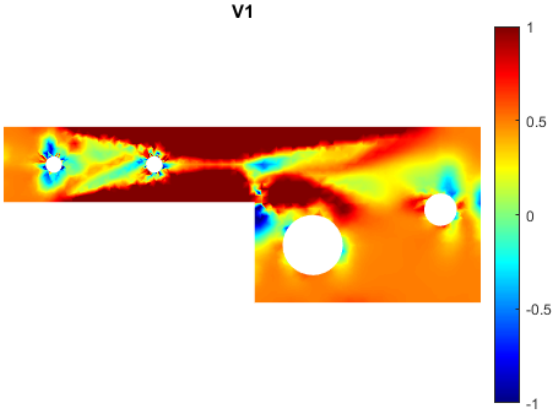
Figure 5.39: Chair bracket: topology optimization, single vs double load case.



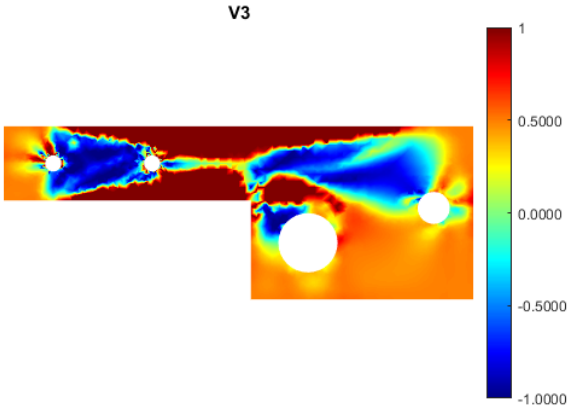
(a) $V_{1\mathbf{A}}$ distribution, load case 1.



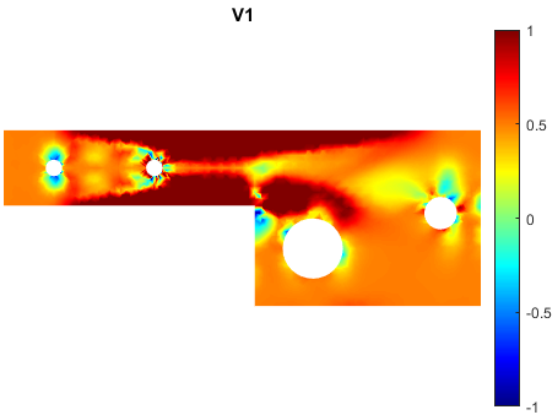
(b) $V_{3\mathbf{A}}$ distribution, load case 1.



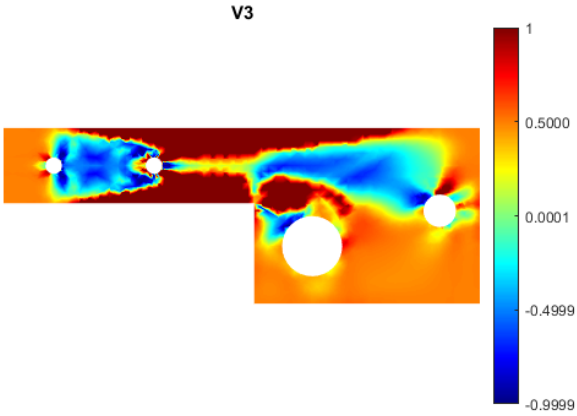
(c) $V_{1\mathbf{A}}$ distribution, double load case.



(d) $V_{3\mathbf{A}}$ distribution, double load case.

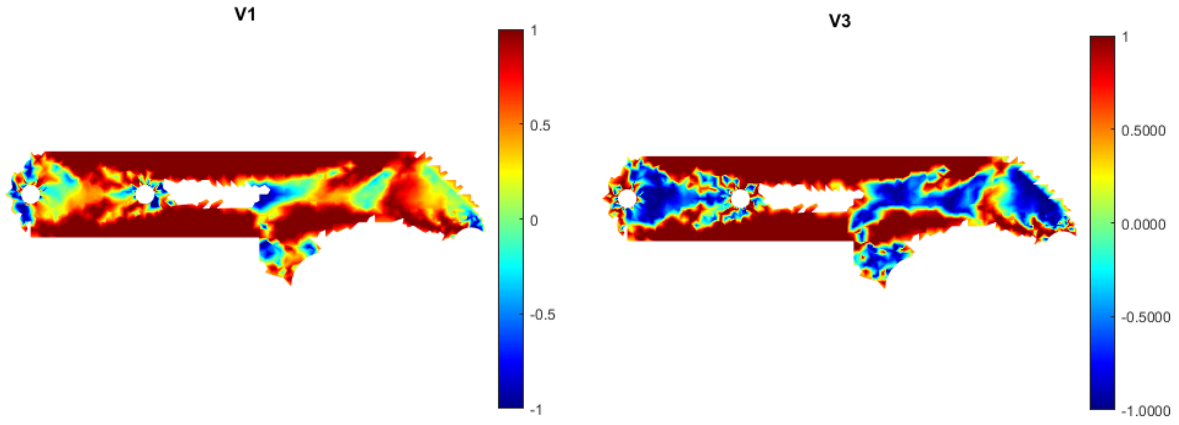


(e) $V_{1\mathbf{A}}$ distribution, load case 2.



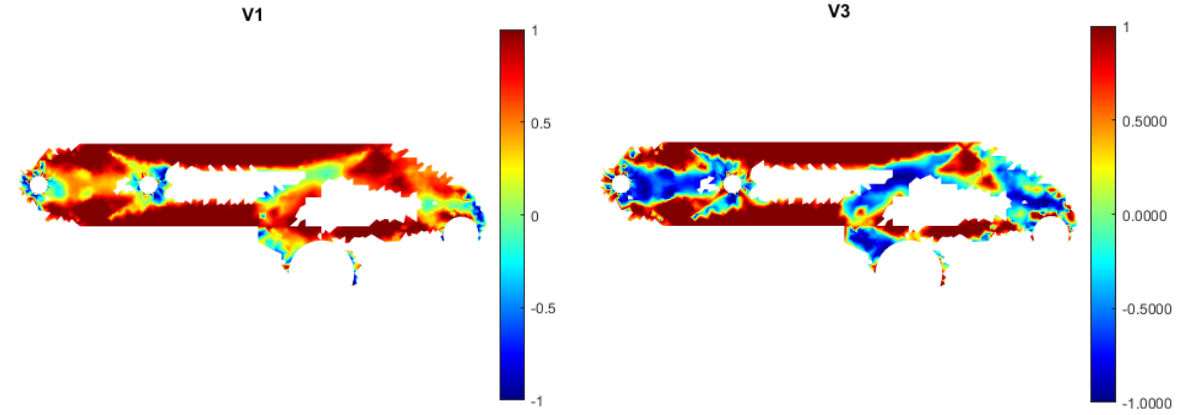
(f) $V_{3\mathbf{A}}$ distribution, load case 2.

Figure 5.40: Chair bracket: variable stiffness design, single vs double load case.



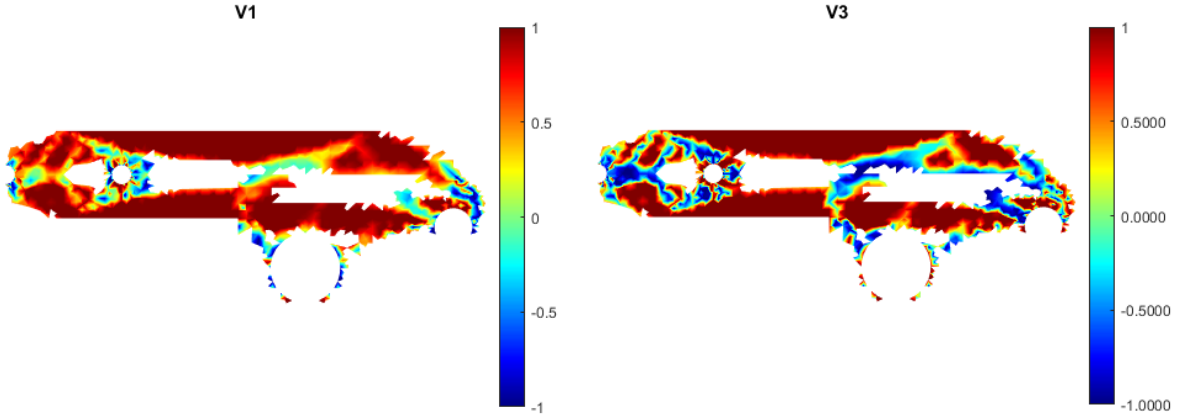
(a) $V_{1\mathbf{A}}$ distribution, load case 1.

(b) $V_{3\mathbf{A}}$ distribution, load case 1.



(c) $V_{1\mathbf{A}}$ distribution, double load case.

(d) $V_{3\mathbf{A}}$ distribution, double load case.



(e) $V_{1\mathbf{A}}$ distribution, load case 2.

(f) $V_{3\mathbf{A}}$ distribution, load case 2.

Figure 5.41: Chair bracket: staggered optimization, single vs double load case.

Regarding topology optimization, load case 1 tends to reinforce the left part of the bracket, where the load is applied on holes 1 and 2, and, while the weights change, this reinforcement shifts to the left side, where the load is applied on holes 3 and 4. It is clear from Figure 5.39 that a certain amount of post-processing needs to be implemented, due to the presence of gray areas in some parts of the designs. As previously mentioned, the absence of a completely black and white solution is attributed to the step size used. The same geometry behavior is noticed with staggered optimization (Figure 5.41). Thicker members are created on the left for load case 1, whereas when the load is shifted to the right, material is "transferred" from the left and placed on the right. In this way, the significance of the loading is emphasized by reinforcing the structure where necessary, depending on the assigned weights. Regarding variable stiffness design (Figure 5.40), most of the changes in the lamination parameter distributions are observed around the area of the holes, as expected.

More emphasis is given on the variation of the fiber orientations around holes 1 and 2 in load case 1, while the intense fiber angles variation is shifted to holes 3 and 4 for load case 2.

The values of compliance at convergence for each example are listed in Table 5.25. It is evident that staggered optimization provides the minimum compliance for all the different scenarios.

	Topology optimization	**Variable stiffness design**	**Staggered optimization**
Compliance [J], load case 1	11.3677	11.2716	6.4079
% change in compliance (w.r.t. topology optimization)	-	-0.8454	-43.6306
Compliance [J], double load case	13.2037	11.4158	7.8405
% change in compliance (w.r.t. topology optimization)	-	-13.5409	-40.6189
Compliance [J], load case 2	14.0828	11.7090	8.0955
% change in compliance (w.r.t. topology optimization)	-	-16.8560	-42.5150

Table 5.25: Optimization results for the chair bracket: single vs double load case.

5.4. Comparison with examples from the literature

Three indicative examples from the literature, one for each optimization method, were used to compare the results obtained from the developed code in this thesis.

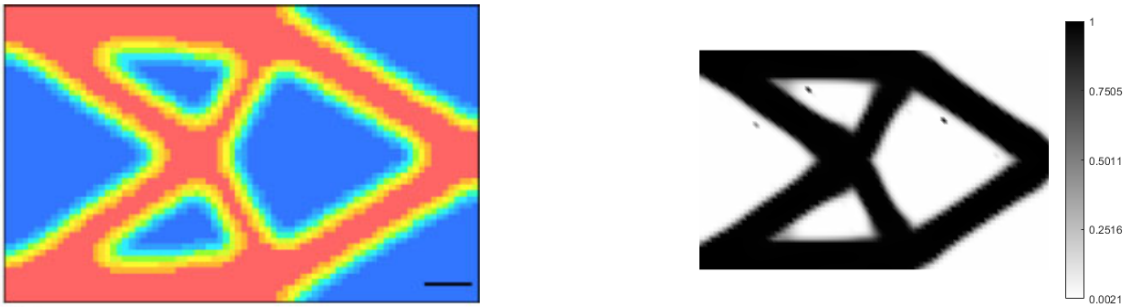
5.4.1. Topology optimization

The first example is a cantilever beam of dimensions 40 mm x 25 mm, as solved in Guest et al. (2004), using an element length of 0.5 mm. The thickness assigned to the structural domain is 1 mm. The boundary conditions applied are identical to Figure 5.1. The point load that the plate is subjected to is equal to 1 N and the material properties used are identical to the ones used in Guest et al. (2004), listed in Table 5.26. The inputs in Guest et al. (2004) are treated as dimensionless and the material used is isotropic.

Material properties	
E_1 [GPa]	1000
E_2 [GPa]	1000
G_{12} [GPa]	400
ν_{12}	0.25

Table 5.26: Material properties used in Guest et al. (2004).

Δp was set to 0.1. The parameters $p_{\max}$, $r_{\min}$ and the volume fraction were set to 5, 2 mm and 0.5, respectively, to match the ones used in Guest et al. (2004). The result of the developed code is illustrated in Figure 5.42b, next to the corresponding result from Guest et al. (2004), in Figure 5.42a. The two solutions are similar. One difference that can be spotted is the presence of gray areas around the structural members in Guest et al. (2004), which does not exist in the solution obtained by this code. On the other hand, irregularities in the solution in Figure 5.42b can be identified, which can be alleviated using a lower $r_{\min}$ or higher $p_{\max}$, as proved in the parametric analysis performed in section §5.1.2.



(a) Result of Guest et al. (2004).

(b) Result of the developed code.

Figure 5.42: Topology optimization; comparison with example from the literature.

5.4.2. Variable stiffness design

The second example is a flat composite plate of dimensions 2000 mm x 400 mm, as solved in Peeters et al. (2018), with a thickness of 1 mm. The boundary conditions applied are illustrated in Figure 5.43. The point load that the plate is subjected to is equal to 10 N and the material properties used are listed in Table 5.27.

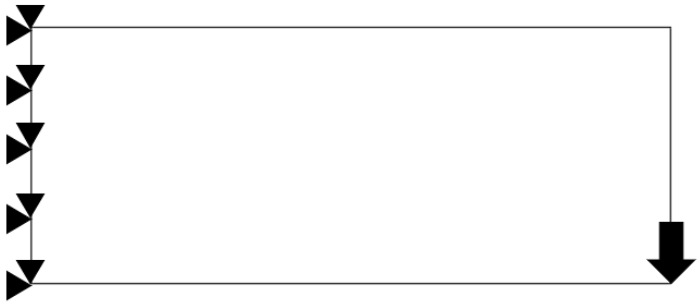
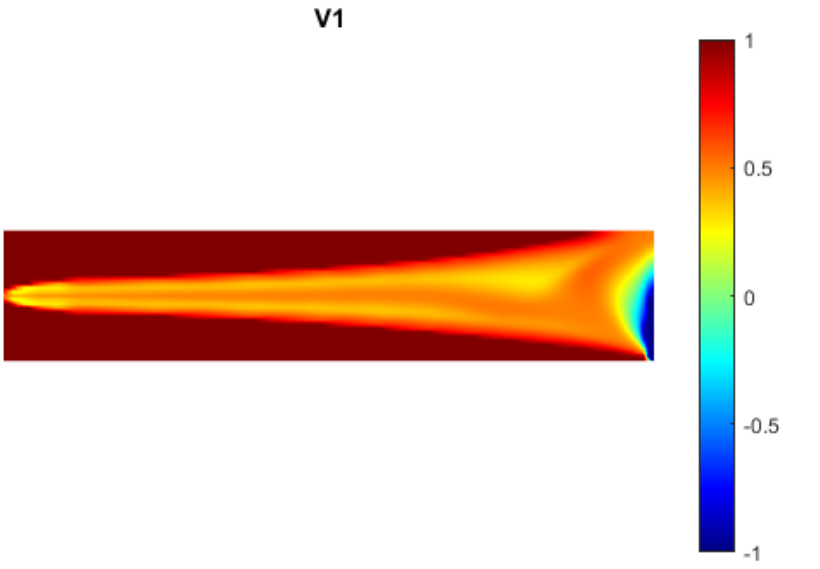


Figure 5.43: Boundary conditions for the literature example, as in Peeters et al. (2018).

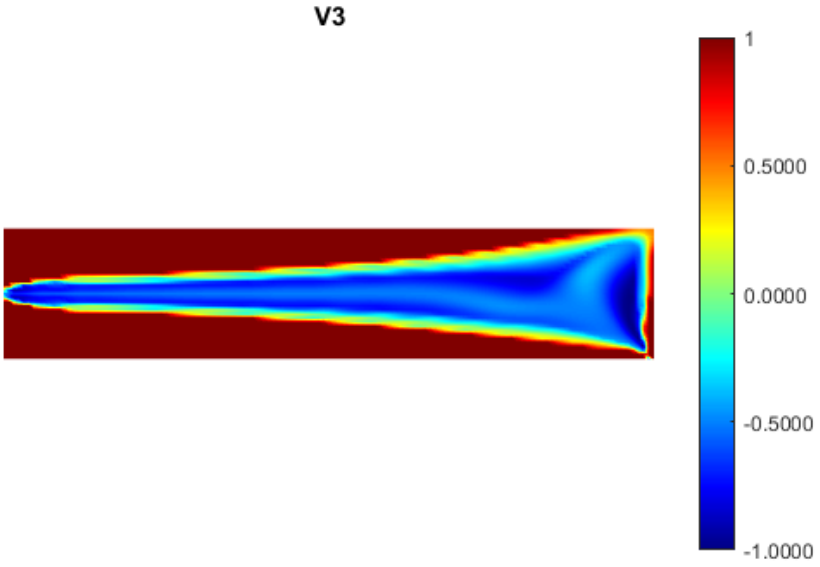
Material properties	
E_1 [GPa]	148
E_2 [GPa]	9.65
G_{12} [GPa]	4.55
ν_{12}	0.3

Table 5.27: Material properties used in Peeters et al. (2018).

123 elements were used along the X direction and 32 along the Y direction, that is double the triangular elements used towards each direction in Peeters et al. (2018). The results of Peeters et al. (2018) are illustrated in Figure 2.15 and the results of the developed code, in Figure 5.44. The distributions obtained are similar. However, more refined details can be observed in Figure 5.44 and this can be attributed to the fact that variable stiffness design is mesh dependent. In this example, a different element type and size was used than in Peeters et al. (2018). Nevertheless, the optimization behaves as expected.



(a) $V_{1\mathbf{A}}$ result of the developed code.



(b) $V_{3\mathbf{A}}$ result of the developed code.

Figure 5.44: Variable stiffness design; comparison with example from the literature.

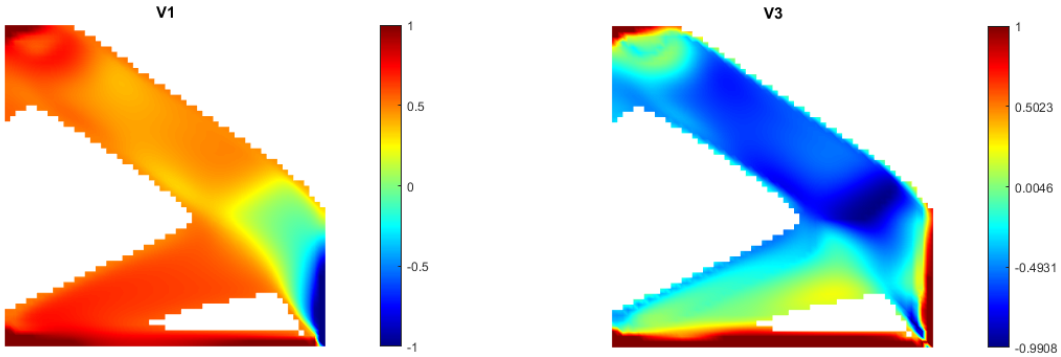
5.4.3. Staggered optimization

The final example is a cantilever beam, as well, as solved in Peeters, van Baalen, et al. (2015). The beam was scaled down to 30 mm x 30 mm instead of 300 mm x 300 mm and solved using an element length of 0.5 mm. The boundary conditions applied on the beam are shown in Figure 2.20. The point load is equal to 10 N and a thickness of 1 mm was assigned to the structure. The material properties used are listed in Table 5.28.

Material properties	
E_1 [GPa]	177
E_2 [GPa]	10.8
G_{12} [GPa]	7.6
ν_{12}	0.27

Table 5.28: Material properties used in Peeters, van Baalen, et al. (2015).

Δp, $p_{\max}$ and $r_{\min}$ were set to 0.1, 4 and 1 mm, respectively. The assigned volume fraction was 0.6, as in Peeters, van Baalen, et al. (2015). The result of Peeters, van Baalen, et al. (2015) is shown in Figure 2.21, whereas the obtained result of the developed code is depicted in Figure 5.45. The designs obtained are also similar in this case. Nevertheless, there are some differences regarding the size of the structural members, in terms of the geometry, and the variation of the lamination parameters at some points, in terms of the stiffness distribution. Those can be attributed to the FE discretization, since a coarser mesh with an element length of 5 mm was used in Peeters, van Baalen, et al. (2015) and also the use of the projection scheme in the developed code, with a $r_{\min}$ equal to 1 mm.



(a) $V_{1\mathbf{A}}$ result of the developed code.

(b) $V_{3\mathbf{A}}$ result of the developed code.

Figure 5.45: Staggered optimization; comparison with example from the literature.

6

Conclusions and Recommendations

Topology optimization is an optimization method capable of producing lightweight structures with optimal performance and has been studied for more than thirty years, primarily for isotropic materials. On the other hand, variable stiffness design is a relatively new research field that owes its birth to the evolution of novel manufacturing techniques. The method is applied on composite structures and its purpose is to fabricate parts with improved structural performance by altering the fiber orientations within each ply.

In this thesis, a MATLAB code was developed to apply the aforementioned optimization methods and the coupling of the two, namely staggered optimization. Three designs were studied in total; a flat composite plate, a lug and an aircraft chair bracket. The meshes of those designs were created on Gmsh and inserted in MATLAB. Then, the MATLAB function `main`, according to the input from the user, was responsible for the pre-processing, optimization and post-processing of the result.

Mesh convergence studies were performed on the plate and the lug for both topology optimization and variable stiffness design. Furthermore, a parametric analysis was presented using the plate example to cover the effect of the different topology optimization parameters on the solution. A number of comparative analyses were performed for the three optimization methods, where the flat composite plate was studied under a single and a double load case, the lug under a single load case and the chair bracket under a single and a double load case and for different volume fractions. These analyses aimed to present the results obtained using each optimization method and quantify their differences in compliance as well as computational time. In the last chapter of the results, indicative examples from the literature were chosen for each optimization method and were compared to the results of the developed code, as an additional form of verification.

6.1. Conclusions

The comparative analyses served as a recap of the thesis and led to a number of valuable findings. In order for the analyses to be valid, the same optimization parameters were used for the three optimization methods. In terms of the designs obtained, it can be concluded that the solution from topology optimization does not always coincide with the solution of the staggered optimization. Since both the densities and the lamination parameters are updated at one iteration during the staggered optimization, the solution is able to pick a different path along the optimization process that can lead to a different geometry. Regarding the compliance obtained, there is no doubt that both variable stiffness design and staggered optimization offer an improved solution compared to topology optimization for a quasi-isotropic material. However, it is observed that, for simple geometries like the flat composite plate, variable stiffness design offers the most efficient structure compared to the other two methods. On the contrary, for relatively more complex examples like the flat composite lug and the aircraft chair bracket, the staggered optimization produces a much stiffer structure than variable stiffness design. The reason for this is that, in case of simpler geometries, variable stiffness design has more freedom to exploit the varying material orientation capabilities, and, thus, is able to perform bet-

ter. Regarding the computational time, when the same optimization parameters are used, variable stiffness design is the most computationally efficient, with topology optimization coming second. The most costly computationally is the staggered optimization. This stems from the fact that topology optimization needs a certain number of iterations for the density penalty power p to reach $p_{\max}$ using the continuation method. This means that the compliance initially increases and, after it reaches $p_{\max}$, it starts dropping steadily. This initial procedure adds a computational cost to the method. Regarding staggered optimization, two optimization methods are taking place and two FE problems are solved in one iteration, making the method the most costly computationally.

Taking into account the conclusions drawn from the comparative analyses, the most efficient method has to be determined based on the application and the available resources. One has to also take into account that variable stiffness design and staggered optimization require two additional steps for the product to be ready for manufacturing; the stacking sequence retrieval and the fiber path construction. Therefore, for a specific design, the advantages and disadvantages for each method have to be estimated in terms of efficiency, cost and computational time.

Moreover, the mesh convergence analysis for topology optimization showed no mesh dependency of the solution and no checkerboarding issues. This is due to the fact that, in this thesis, nodal values were used as design parameters instead of element values and a density filtering, namely projection scheme, was implemented. In addition, the continuation method was used to ensure that the solution does not fall into a local minimum. On the contrary, the result of the variable stiffness design using lamination parameters got more detailed with mesh refinement. This mesh dependency of the method is due to the fact that no filtering scheme or curvature constraint was implemented for the lamination parameters.

Finally, certain conclusions can be drawn from the parametric analysis on topology optimization, as well. Increasing the maximum density penalty power $p_{\max}$, a more intense penalization of the densities takes place and the solution becomes more well defined towards the desirable black and white one. The same effect is achieved by decreasing the minimum radius $r_{\min}$ for the projection scheme. In this case, the eligible area taken into account for the calculation of the element density becomes smaller, and thus, the eligible nodes become less, leading to a more distinct solution. The continuation method step size Δp does not influence the solution; the method makes it possible to avoid local minima irrespective of the step size used.

6.2. Recommendations

The developed code offers the possibility to optimize a custom design using one of the following three optimization methods; topology optimization, variable stiffness design and staggered optimization. There is, of course, room for improvement and upscale possibilities that can serve as a follow-up of this work.

First of all, variable stiffness design can be fully implemented using the three-step approach (IJsselmuiden, 2011), including the steps of the stacking sequence retrieval and fiber path construction, in order for the result to be manufacturable. A desirable characteristic would be to control the fiber angle orientations, preventing them to turn abruptly, in order to obtain a smooth fiber angle distribution for manufacturing. This has been previously studied in the literature and it is known as the curvature constraint, imposing a threshold on the turning radius according to the limitations of the AFP machine. Another interesting investigation would be to implement a filter on the lamination parameters, similar to the projection scheme used for topology optimization (Guest et al., 2004) and study the effect on the mesh dependency observed with the laminate stiffness optimization in this work. Moreover, this result could be compared with the implementation of the curvature constraint using the direct, indirect and hybrid methods presented in Hong et al. (2020).

A

Gmsh .geo file

In this Appendix chapter a short version of the .geo file for the flat composite lug is presented, in order to explain the key points of the file structure.

```
// Flat composite lug .geo file

// Set geometry kernel
SetFactory("OpenCASCADE");

// Set variables
length = 533.6;
height = 175;
centre2centre = 310;
centre2side = 111.8;
centre2top = 87.5;
fillet = 50;
diameter = 86;

// Create points
Point(1) = {0,0,0};
Point(2) = {length,0,0};
Point(3) = {0,height,0};
Point(4) = {length,height,0};
...

// Create circular arcs
Circle(1) = {5,7,6};
Circle(2) = {8,10,9};
Circle(3) = {11,13,12};
Circle(4) = {14,16,15};

// Create lines
Line(5) = {6,9};
Line(6) = {8,11};
Line(7) = {12,15};
Line(8) = {14,5};
...
```

```
// Create points and arcs for hole #1
Point(19) = {centre2side-diameter/2,height/2,0};
Point(20) = {centre2side-(diameter/2)*Cos(Pi/4),centre2top+(diameter/2)
*Sin(Pi/4),0};
Point(21) = {centre2side-(diameter/2)*Cos(2*Pi/4),centre2top+(diameter/2)
*Sin(2*Pi/4),0};
Point(22) = {centre2side-(diameter/2)*Cos(3*Pi/4),centre2top+(diameter/2)
*Sin(3*Pi/4),0};
Point(23) = {centre2side-(diameter/2)*Cos(Pi),centre2top+(diameter/2)
*Sin(Pi),0};
Point(24) = {centre2side-(diameter/2)*Cos(5*Pi/4),centre2top+(diameter/2)
*Sin(5*Pi/4),0};
Point(25) = {centre2side-(diameter/2)*Cos(6*Pi/4),centre2top+(diameter/2)
*Sin(6*Pi/4),0};
Point(26) = {centre2side-(diameter/2)*Cos(7*Pi/4),centre2top+(diameter/2)
*Sin(7*Pi/4),0};
Circle(9) = {19,17,20};
Circle(10) = {20,17,21};
Circle(11) = {21,17,22};
Circle(12) = {22,17,23};
Circle(13) = {23,17,24};
Circle(14) = {24,17,25};
Circle(15) = {25,17,26};
Circle(16) = {26,17,19};
Line(17) = {3,20};
Point(28) = {2*centre2side, height,0};
Point(29) = {2*centre2side, 0,0};
...

// Create points and arcs for hole #2
...

// Compute fragments between entities and delete the initial entities
BooleanFragments{Curve{5};Delete;}{Curve{24};Curve{38};Delete;}
BooleanFragments{Curve{7};Delete;}{Curve{38};Curve{22};Delete;}
...

// Define sequences of closed curves and the respective surfaces
Curve Loop(1) = {49, -9, 55, -61, 46};
Plane Surface(1) = {1};
Curve Loop(2) = {49, 10, -54, 68, -47};
Plane Surface(2) = {2};
...

// Apply the transfinite meshing constraint to construct a structured grid
Transfinite Surface {1};
Transfinite Surface {2};
...

// Apply a certain number of equidistant nodes at each curve
Transfinite Curve {43, 60, 40} = 25 Using Progression 1;
...

// Convert triangular to quadrilateral elements
Recombine Surface {10};
...
```

B

Gmsh .msh file

In this Appendix chapter the .msh file structure for mesh 1 used in the convergence analysis for the flat composite lug is presented.

```
$MeshFormat // Initializes file format section
2.2 0 8 // Version number, file type (0 for ASCII mode), data size
$EndMeshFormat // Ends file format section
$Nodes // Initializes node section
778 // Number of nodes
1 0 50 0 // Node ID - X coord - Y coord - Z coord
2 50 0 0
3 0 125 0
4 50 175 0
...
$EndNodes // Ends node section
$Elements // Initializes element section
969 // Number of elements
1 15 2 0 5 1 // Element ID - 1 node point - Number of tags - Tag 1 - Tag 2 - Node
2 15 2 0 6 2
3 15 2 0 14 3
4 15 2 0 15 4
...
25 1 2 0 9 6 25 // Element ID - 2 node line - Number of tags - Tag 1 - Tag 2 - Node 1
- Node 2
26 1 2 0 9 25 26
27 1 2 0 9 26 27
28 1 2 0 9 27 28
...
268 3 2 0 1 6 25 235 147 // Element ID - 4 node quadrangle - Number of tags - Tag 1
- Tag 2 - Node 1 - Node 2 - Node 3 - Node 4
269 3 2 0 1 25 26 236 235
270 3 2 0 1 26 27 237 236
271 3 2 0 1 27 28 238 237
...
$EndElements // Ends element section
```

C

Extensive function inputs / outputs

In Tables C.1 and C.2, whenever "iteration" is mentioned, it is referred to the topology optimization outer loop iteration, whereas in Tables C.3, C.4 and C.5 it is referred to the variable stiffness design loop iteration.

Input	Description
rho	Vector containing the density nodal values ρ_j at previous iteration
V1	Vector containing the lamination parameter nodal values $V_{1\mathbf{A}j}$ at previous iteration
V3	Vector containing the lamination parameter nodal values $V_{3\mathbf{A}j}$ at previous iteration
topology_loop	See Table 4.29
compliance	Compliance value at previous iteration
NodeCoords	See Table 4.2
NumOfNodes	—"—
NumOfElements	—"—
cnc	—"—
Strains	Strains matrix at previous iteration
Stresses	Stresses matrix at previous iteration
p	Density penalty power
G0	Matrix $\mathbf{\Gamma_0}$
G1	Matrix $\mathbf{\Gamma_1}$
G3	Matrix $\mathbf{\Gamma_3}$
PowerAddition_TO_step	See Table 4.29
PowerAddition_TO_lambda	See table 4.31
step_TO	Step size at previous iteration, see Table 4.29
AllDOF	See Table 4.2
ksi_s	—"—
eta_s	—"—
RestrainedDOF	—"—
ForcesVector	—"—
h	Laminate thickness
p_vector_TO	Vector containing the density penalty powers at each iteration
step_vector_TO	Vector containing the topology optimization step sizes at each iteration
compliance_change_vector_TO	Vector containing the compliance convergence fractions at each iteration
compliance_vector_TO	Vector containing the compliance values at each iteration
rho_matrix	Matrix; each row corresponds to an iteration and each column to the density nodal values ρ_j
stresses_matrix_TO	Matrix containing the stresses at each iteration
strains_matrix_TO	Matrix containing the strains at each iteration
displacements_matrix_TO	Matrix containing the displacements at each iteration
NeighboringElements	See Table 4.27
rmin	See Table 4.15
ProjectedNodes	See Table 4.17
Distance	—"—
lambda_loop	Number of previous inner loop iterations
lambda_storage	Structure; each row corresponds to an outer loop iteration and each column to the Lagrange multiplier values at each inner loop iteration
lambda_temp	Lagrange multiplier Λ value at previous iteration
resource	See Table 4.31
p_init	See Table 4.29
p_max	—"—
p_incr	—"—

Table C.1: Input of function `TO`.

Output	Description
rho	Vector containing the density nodal values ρ_j at current iteration
lambda_temp	Lagrange multiplier Λ value at current iteration
topology_loop	See Table 4.29
compliance	Compliance value at current iteration
Strains	Strains matrix at current iteration
Stresses	Stresses matrix at current iteration
p	Density penalty power
p_vector_TO	Vector containing the density penalty powers at each iteration
step_vector_TO	Vector containing the topology optimization step sizes at each iteration
compliance_change_TO	Compliance convergence fraction at current iteration
compliance_change_vector_TO	Vector containing the compliance convergence fractions at each iteration
compliance_vector_TO	Vector containing the compliance values at each iteration
rho_matrix	Matrix; each row corresponds to an iteration and each column to the density nodal values ρ_j
stresses_matrix_TO	Matrix containing the stresses at each iteration
strains_matrix_TO	Matrix containing the strains at each iteration
lambda_storage	Structure; each row corresponds to an outer loop iteration and each column to the Lagrange multiplier values at each inner loop iteration
step_TO	Step size at current iteration
lambda_loop	Number of current inner loop iterations
displacements_matrix_TO	Matrix containing the displacements at each iteration

Table C.2: Output of function `TO`.

Input	Description
rho	Vector containing the density nodal values ρ_j at previous iteration
V1	Vector containing the lamination parameter nodal values $V_{1\mathbf{A}j}$ at previous iteration
V3	Vector containing the lamination parameter nodal values $V_{3\mathbf{A}j}$ at previous iteration
varstiff_loop	See Table 4.35
compliance	Compliance value at previous iteration
NodeCoords	See Table 4.2
NumOfNodes	—"—
NumOfElements	—"—
cnc	—"—
Strains	Strains matrix at previous iteration
Stresses	Stresses matrix at previous iteration
p	Density penalty power
G0	Matrix $\mathbf{\Gamma_0}$
G1	Matrix $\mathbf{\Gamma_1}$
G3	Matrix $\mathbf{\Gamma_3}$
PowerAddition_VS_step	See Table 4.35
step1	—"—
step2	—"—
AllDOF	See Table 4.2
ksi_s	—"—
eta_s	—"—
RestrainedDOF	—"—
ForcesVector	—"—
h	Laminate thickness
p_vector_VS	Vector containing the density penalty powers at each iteration
step_vector_VS	Vector containing the variable stiffness design step sizes at each iteration
compliance_change_vector_VS	Vector containing the compliance convergence fractions at each iteration
compliance_vector_VS	Vector containing the compliance values at each iteration
V_matrix	Matrix; every two rows correspond to an iteration, each odd's row column to the lamination parameter nodal values $V_{1\mathbf{A}j}$ and each even's row column to the lamination parameter nodal values $V_{3\mathbf{A}j}$
stresses_matrix_VS	Matrix containing the stresses at each iteration
strains_matrix_VS	Matrix containing the strains at each iteration
displacements_matrix_VS	Matrix containing the displacements at each iteration
NeighboringElements	See Table 4.27

Table C.3: Input of function `VSonly`.

Input	Description
rho	Vector containing the density nodal values ρ_j at previous iteration
V1	Vector containing the lamination parameter nodal values $V_{1\mathbf{A}j}$ at previous iteration
V3	Vector containing the lamination parameter nodal values $V_{3\mathbf{A}j}$ at previous iteration
varstiff_loop	See Table 4.35
compliance	Compliance value at previous iteration
NodeCoords	See Table 4.2
NumOfNodes	—”—
NumOfElements	—”—
cnc	—”—
Strains	Strains matrix at previous iteration
Stresses	Stresses matrix at previous iteration
p	Density penalty power
G0	Matrix $\mathbf{\Gamma_0}$
G1	Matrix $\mathbf{\Gamma_1}$
G3	Matrix $\mathbf{\Gamma_3}$
PowerAddition_VS_step	See Table 4.35
step1	—”—
step2	—”—
AllDOF	See Table 4.2
ksi_s	—”—
eta_s	—”—
RestrainedDOF	—”—
ForcesVector	—”—
h	Laminate thickness
p_vector_VS	Vector containing the density penalty powers at each iteration
step_vector_VS	Vector containing the variable stiffness desigb step sizes at each iteration
compliance_change_vector_VS	Vector containing the compliance convergence fractions at each iteration
compliance_vector_VS	Vector containing the compliance values at each iteration
V_matrix	Matrix; every two rows correspond to an iteration, each odd's row column to the lamination parameter nodal values $V_{1\mathbf{A},j}$ and each even's row column to the lamination parameter nodal values $V_{3\mathbf{A},j}$
stresses_matrix_VS	Matrix containing the stresses at each iteration
strains_matrix_VS	Matrix containing the strains at each iteration
displacements_matrix_VS	Matrix containing the displacements at each iteration
NeighboringElements	See Table 4.27
rmin	See Table 4.15
ProjectedNodes	See Table 4.17
Distance	—”—

Table C.4: Input of function `VS`.

Output	**Description**
V1	Vector containing the lamination parameter nodal values $V_{1\mathbf{A}j}$ at current iteration
V3	Vector containing the lamination parameter nodal values $V_{3\mathbf{A}j}$ at current iteration
varstiff_loop	See Table 4.35
compliance	Compliance value at current iteration
Strains	Strains matrix at current iteration
Stresses	Stresses matrix at current iteration
p_vector_VS	Vector containing the density penalty powers at each iteration
step_vector_VS	Vector containing the variable stiffness step sizes at each iteration
compliance_change_VS	Compliance convergence fraction at current iteration
compliance_change_vector_VS	Vector containing the compliance convergence fractions at each iteration
compliance_vector_VS	Vector containing the compliance values at each iteration
V_matrix	Matrix; every two rows correspond to an iteration, each odd's row column to the lamination parameter nodal values $V_{1\mathbf{A}j}$ and each even's row column to the lamination parameter nodal values $V_{3\mathbf{A}j}$
stresses_matrix_VS	Matrix containing the stresses at each iteration
strains_matrix_VS	Matrix containing the strains at each iteration
step1	Step size at current iteration, see Table 4.35
step2	Step size at current iteration, see Table 4.35
displacements_matrix_VS	Matrix containing the displacements at each iteration

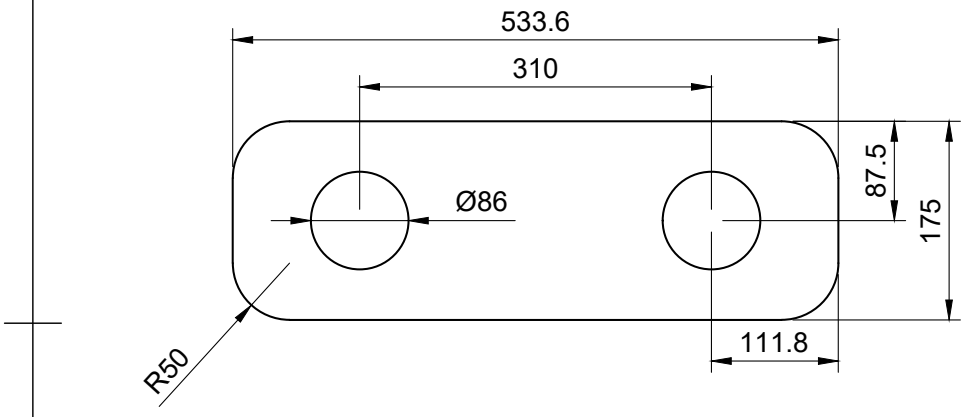
Table C.5: Output of functions `VSonly` and `VS`.

Input	**Description**
optimization_opt	Optimization method switch
topology_loop	Number of topology optimization iterations
varstiff_loop	Number of variable stiffness design iterations
compliance_vector_TO	Vector containing the compliance values from topology optimization
compliance_vector_VS	Vector containing the compliance values from variable stiffness design
rho_vector	Vector containing the updated density nodal values ρ_j at each topology optimization iteration
V_vector	Vector containing the updated lamination parameter nodal values $V_{1\mathbf{A}j}$, $V_{3\mathbf{A}j}$ at each variable stiffness design iteration
cnc	See Table 4.2
NodeCoords	—"—
compliance_change_vector_TO	Vector containing the compliance convergence fractions for topology optimization
compliance_change_vector_VS	Vector containing the compliance convergence fractions for variable stiffness design
times	Vector containing the wall-clock time at the end of each iteration
times_CPU	Vector containing the CPU time at the end of each iteration

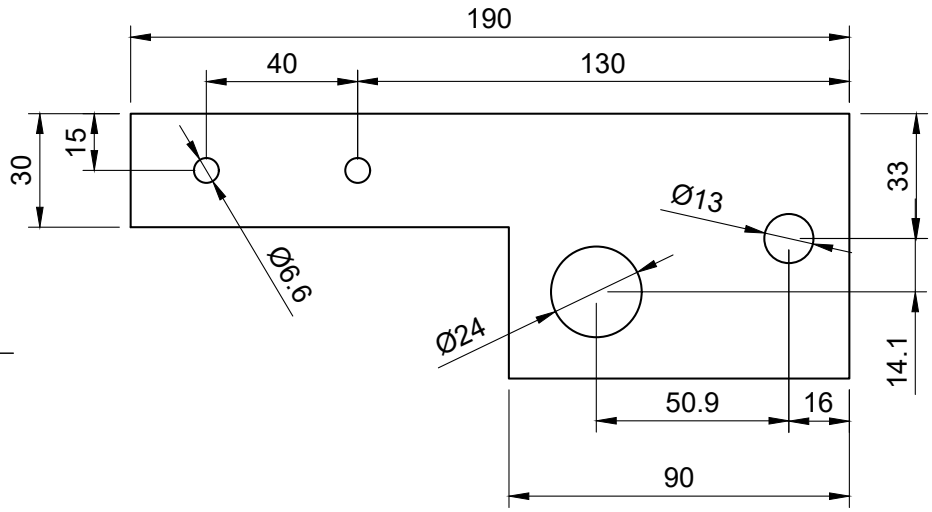
Table C.6: Input of function `PlotOptimizationResults`.

D

Design dimensions



Dept.	Technical reference	Created by	Approved by
ASM		Nikoleta Pasvanti 18/06/2021	
		Document type	Document status
		Title: Flat composite lug	DWG No.
			Rev. / Date of issue / Sheet: 1/1



Dept.	Technical reference	Created by	Approved by
ASM		Nikoleta Pasvanti 18/06/2021	
		Document type	Document status
		Title: Chair Bracket	DWG No.
			Rev. / Date of issue / Sheet: 1/1

E

Compliance vs number of iterations graphs

During the optimization process, a certain number of iterations are needed for the compliance to converge to a minimum value. The number of iterations required for convergence depends primarily on the step size used, which, in this thesis, is calibrated by the user.

The graphs of the compliance versus the number of iterations for the parametric analysis for the case of the flat composite plate, analyzed in section §5.1.2, are depicted in Figures E.1 to E.12. The compliance value initially increases and starts dropping afterwards. This happens due to the continuation method for the density penalty power p; the optimization starts with an initial value of 1 for p and gradually increases to $p_{\max}$ and, thus, the compliance increases as well. After p reaches the maximum value, the compliance starts decreasing until it reaches a minimum.

The graphs for the flat composite plate under a single and a double load case, solved in §5.1.3.1 and §5.1.3.2, respectively, are shown in Figures E.13 to E.18. For topology and staggered optimization, the behavior that is observed is the one explained in the previous paragraph. This is not the case for variable stiffness design, where the compliance reduces from the beginning of the optimization process.

Figure E.15 illustrates the compliance after both topology optimization and variable stiffness design. Since small fixed step sizes are used for both and variable stiffness design follows after topology optimization at each iteration, the two compliance values at one iteration are very close to each other. That is the reason why the graph gives the impression that the values coincide.

For the double load case, three graphs are plotted for each optimization method. These represent the compliance of the first and the second load case, as well as the weighted compliance of the two, as calculated by (4.5.4). For the staggered optimization of the double load case, the compliance values obtained from topology optimization and variable stiffness design at one iteration are plotted with the same color for simplicity.

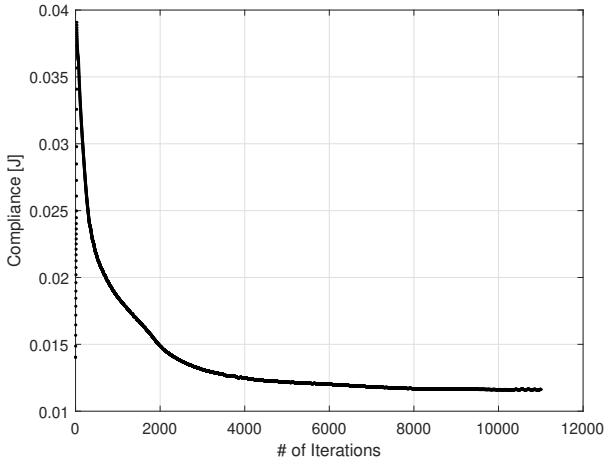


Figure E.1: Plate parametric analysis, r_{min} = 0.5 mm: compliance vs number of topology optimization iterations.

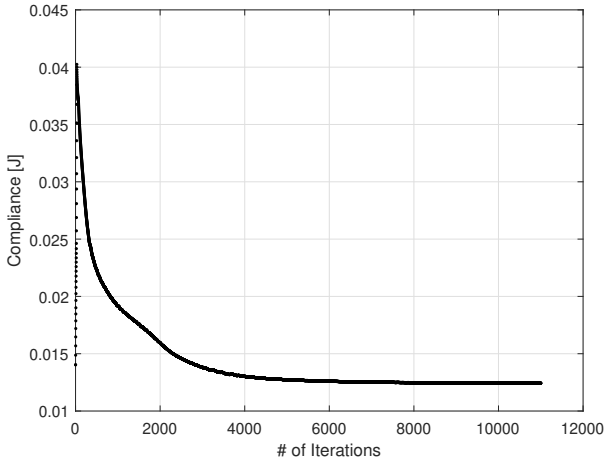


Figure E.2: Plate parametric analysis, r_{min} = 1 mm: compliance vs number of topology optimization iterations.

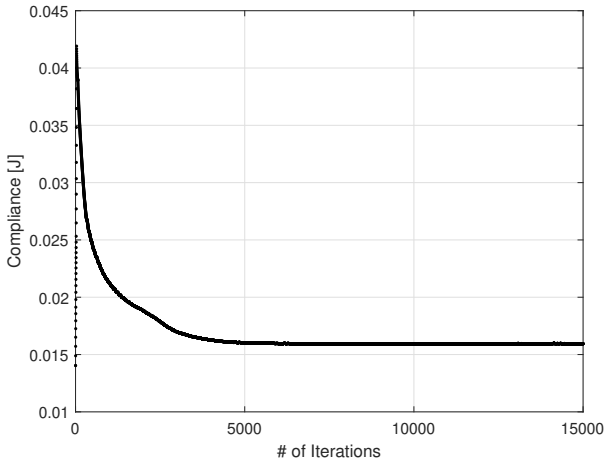


Figure E.3: Plate parametric analysis, r_{min} = 2 mm: compliance vs number of topology optimization iterations.

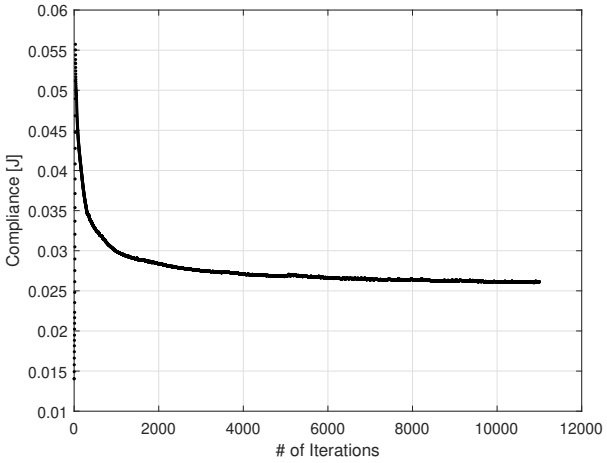


Figure E.4: Plate parametric analysis, $r_{\min} = 4$ mm: compliance vs number of topology optimization iterations.

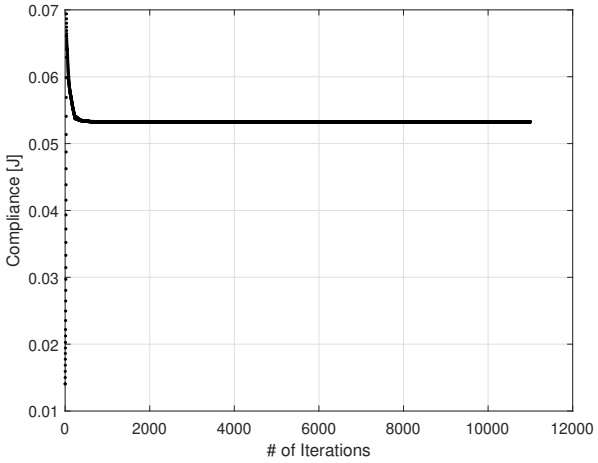


Figure E.5: Plate parametric analysis, $r_{\min} = 8$ mm: compliance vs number of topology optimization iterations.

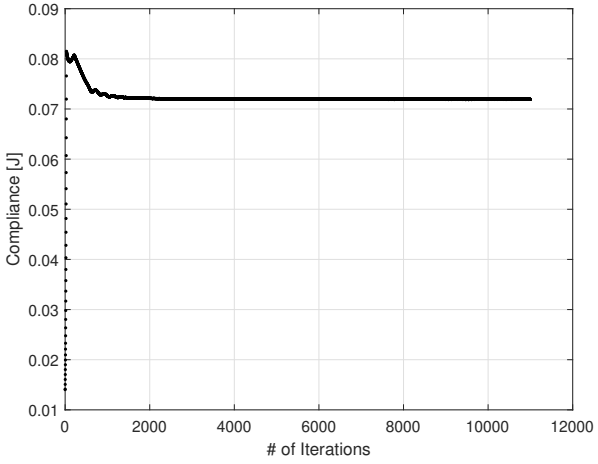


Figure E.6: Plate parametric analysis, $r_{\min} = 12$ mm: compliance vs number of topology optimization iterations.

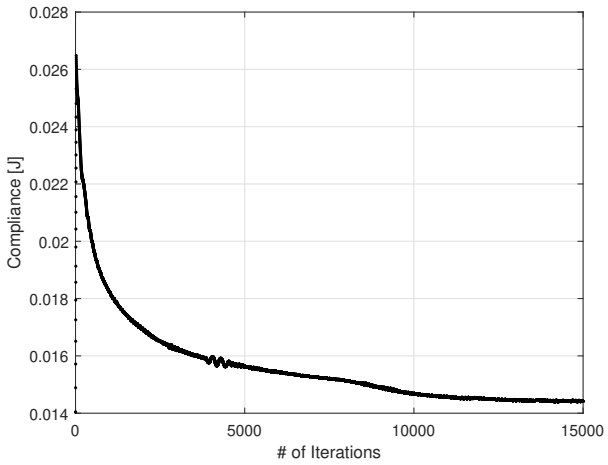


Figure E.7: Plate parametric analysis, $p_{\max} = 3$: compliance vs number of topology optimization iterations.

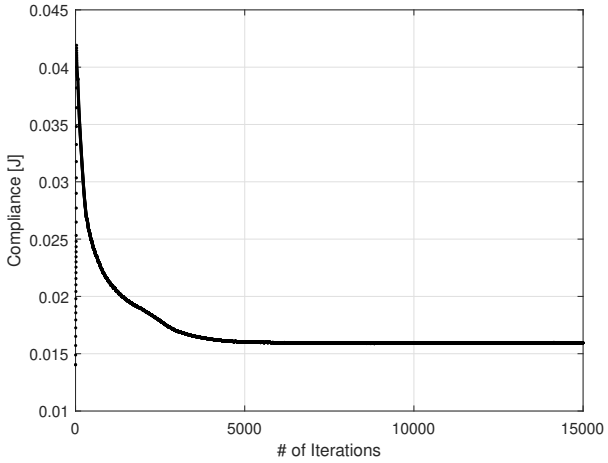


Figure E.8: Plate parametric analysis, $p_{\max} = 4$: compliance vs number of topology optimization iterations.

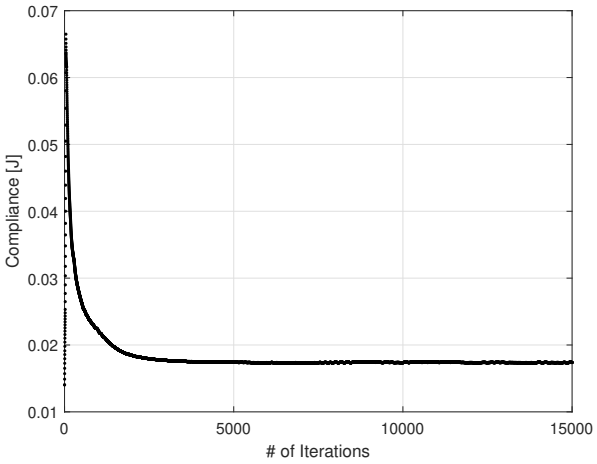


Figure E.9: Plate parametric analysis, $p_{\max} = 5$: compliance vs number of topology optimization iterations.

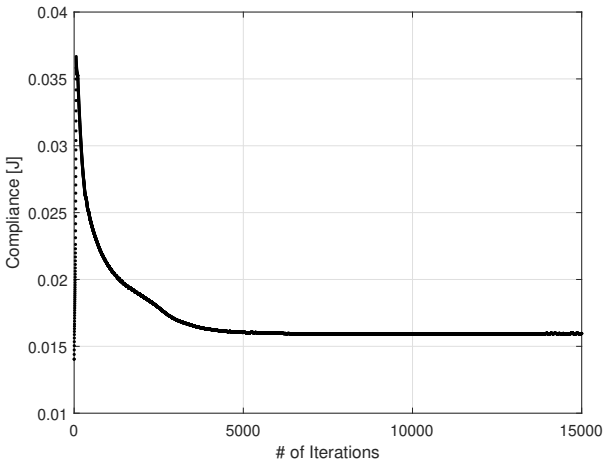


Figure E.10: Plate parametric analysis, $\Delta p = 0.05$: compliance vs number of topology optimization iterations.

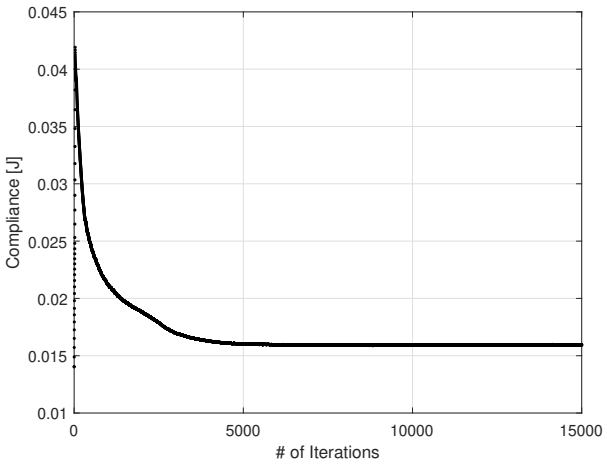


Figure E.11: Plate parametric analysis, $\Delta p = 0.1$: compliance vs number of topology optimization iterations.

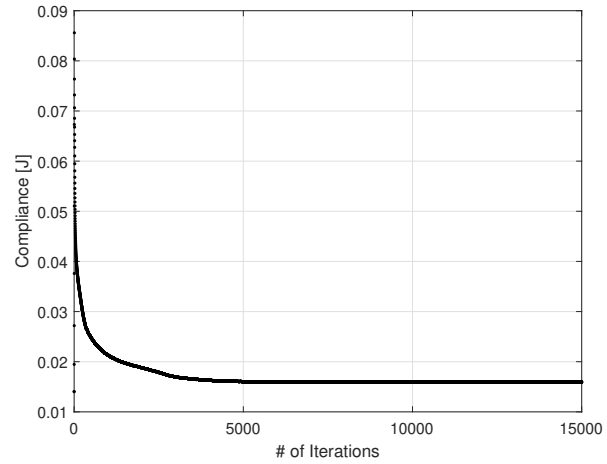


Figure E.12: Plate parametric analysis, $\Delta p = 0.5$: compliance vs number of topology optimization iterations.

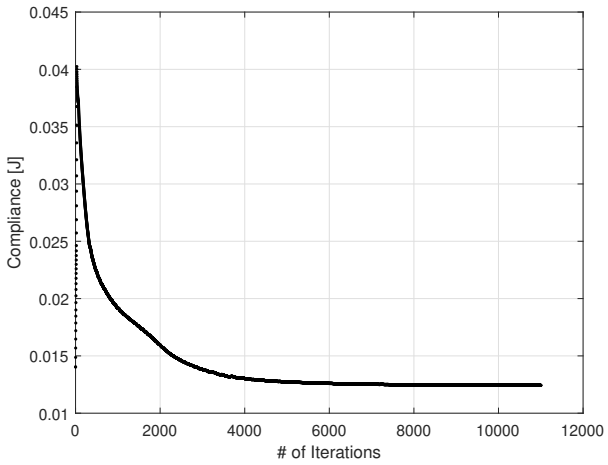


Figure E.13: Flat composite plate, single load case: compliance vs number of topology optimization iterations.

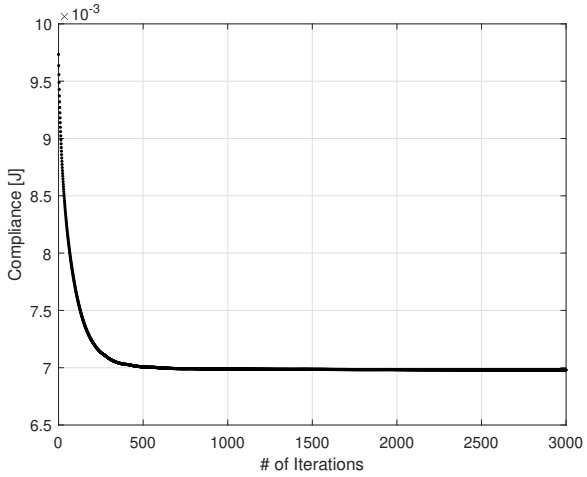


Figure E.14: Flat composite plate, single load case: compliance vs number of variable stiffness design iterations.

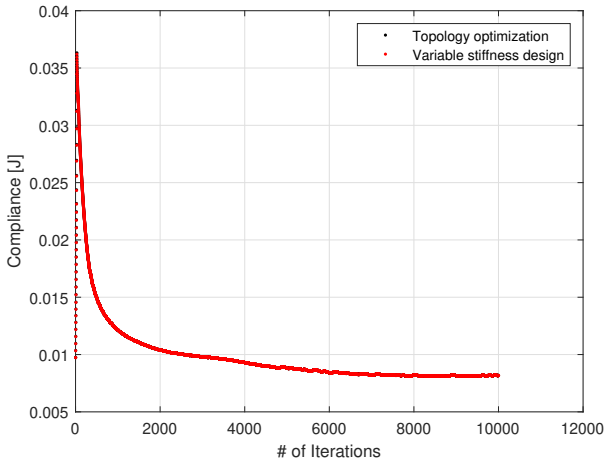


Figure E.15: Flat composite plate, single load case: compliance vs number of staggered optimization iterations.

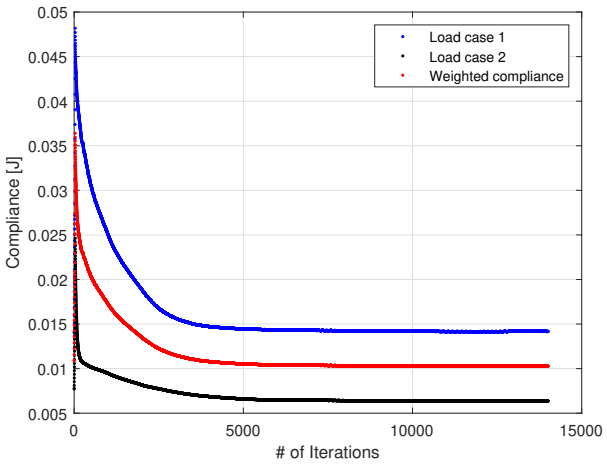


Figure E.16: Flat composite plate, double load case: compliance vs number of topology optimization iterations.

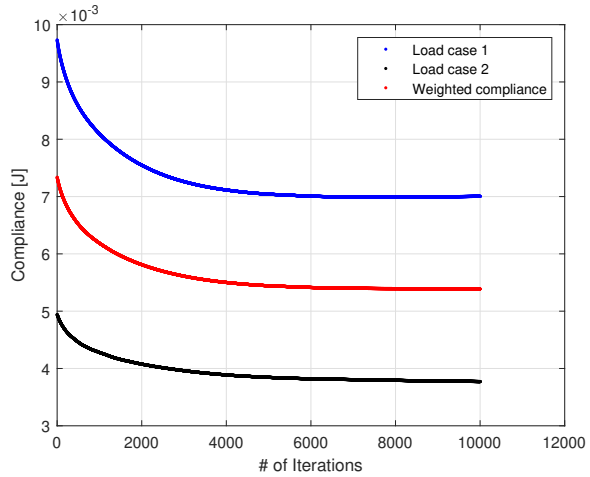


Figure E.17: Flat composite plate, double load case: compliance vs number of variable stiffness design iterations.

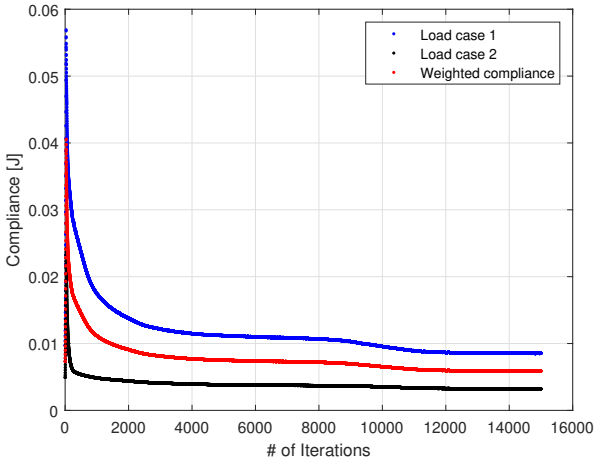


Figure E.18: Flat composite plate, double load case: compliance vs number of staggered optimization iterations.

Bibliography

Abdalla, M., Kassapoglou, C., & Gürdal, Z. (2009). Formulation of composite laminate robustness constraint in lamination parameters space. *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, (May), 1–15. https://doi.org/10.2514/6.2009-2478

Abdalla, M., Setoodeh, S., & Gürdal, Z. (2007). Design of variable stiffness composite panels for maximum fundamental frequency using lamination parameters. *Composite Structures*, *81*(2), 283–291. https://doi.org/10.1016/j.compstruct.2006.08.018

Albazzan, M., Harik, R., Tatting, B., & Gürdal, Z. (2019). Efficient design optimization of nonconventional laminated composites using lamination parameters: A state of the art. *Composite Structures*, *209*, 362–374. https://doi.org/10.1016/j.compstruct.2018.10.095

Albazzan, M., Harik, R., Tatting, B. F., Gürdal, Z., Blom-Schieber, A. W., Rassaian, M., & Wanthal, S. P. (2018). Optimization of cylinders with holes under bending using nonconventional laminates. *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2018*, (210049), 1–17. https://doi.org/10.2514/6.2018-1377

Allaire, G., Jouve, F., & Toader, A.-M. (2004). Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, *194*(1), 363–393. https://doi.org/10.1016/j.jcp.2003.09.032

Allaire, G., Jouve, F., & Toader, A.-M. (2002). A level-set method for shape optimization. *Comptes Rendus Mathematique*, *334*(12), 1125–1130. https://doi.org/https://doi.org/10.1016/S1631-073X(02)02412-3

Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B., & Sigmund, O. (2011). Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization*, *43*(1), 1–16. https://doi.org/10.1007/s00158-010-0594-7

Arora, J. S. (2012). Numerical Methods for Unconstrained Optimum Design. *Introduction to optimum design* (pp. 411–441). Elsevier. https://doi.org/10.1016/b978-0-12-381375-6.00010-3

Bendsøe, M. (1989). Optimal shape design as a material distribution problem. *Structural Optimization*, *1*(4), 193–202. https://doi.org/10.1007/BF01650949

Bendsøe, M., & Kikuchi, N. (1988). Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, *71*, 197–224.

Bendsøe, M., & Sigmund, O. (1999). Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, *69*(9-10), 635–654. https://doi.org/10.1007/s004190050248

Bendsøe, M., & Sigmund, O. (2004). *Topology Optimization*. https://doi.org/10.1007/978-3-662-05086-6

Blom, A., Abdalla, M., & Gürdal, Z. (2010). Optimization of course locations in fiber-placed panels for general fiber angle distributions. *Composites Science and Technology*, *70*(4), 564–570. https://doi.org/10.1016/j.compscitech.2009.12.003

Bloomfield, M. W., Diaconu, C. G., & Weaver, P. M. (2009). On feasible regions of lamination parameters for lay-up optimization of laminated composites. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *465*(2104), 1123–1143. https://doi.org/10.1098/rspa.2008.0380

Bourdin, B. (2001). Filters in topology optimization. *International Journal for Numerical Methods in Engineering*, *50*(9), 2143–2158. https://doi.org/10.1002/nme.116

Boyd, S., & Vandenberghe, L. (2004). *Convex Optimization*. https://doi.org/10.1017/cbo9780511804441

Bruns, T., & Tortorelli, D. (2001). Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, *190*(26-27), 3443–3459. https://doi.org/10.1016/S0045-7825(00)00278-4

Burkardt, J. (2019). gmsh_io, a MATLAB code which can read and write the files used by the GMSH meshing program. https://people.sc.fsu.edu/~jburkardt/m_src/gmsh_io/gmsh_io.html

Díaz, A., & Sigmund, O. (1995). Checkerboard patterns in layout optimization. *Structural Optimization*, *10*(1), 40–45. https://doi.org/10.1007/BF01743693

Geuzaine, C., & Remacle, J.-F. (2009). Gmsh: A 3-D Finite Element Mesh Generator with Built-in Pre- and Post-Processing Facilities. *International Journal for Numerical Methods in Engineering*, *79*, 1309–1331. https://doi.org/10.1002/nme.2579

Guest, J., Prévost, J., & Belytschko, T. (2004). Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International Journal for Numerical Methods in Engineering*, *61*(2), 238–254. https://doi.org/10.1002/nme.1064

Haftka, R. T., & Gürdal., Z. (1993). *Elements of Structural Optimization* (3rd ed.). Springer Netherlands. https://doi.org/10.1007/978-94-011-2550-5

Hammer, V. B., Bendsøe, M. P., Lipton, R., & Pedersen, P. (1997). Parametrization in laminate design for optimal compliance. *International Journal of Solids and Structures*, *34*(4), 415–434. https://doi.org/10.1016/s0020-7683(96)00023-6

Hassani, B., Khanzadi, M., & Tavakkoli, S. (2012). An isogeometrical approach to structural topology optimization by optimality criteria. *Structural and Multidisciplinary Optimization*, *45*(2), 223–233. https://doi.org/10.1007/s00158-011-0680-5

Hong, Z., Peeters, D., & Turteltaub, S. (2020). An enhanced curvature-constrained design method for manufacturable variable stiffness composite laminates. *Computers and Structures*, *238*, 106284. https://doi.org/10.1016/j.compstruc.2020.106284

IJsselmuiden, S. (2011). *Optimal design of variable stiffness composite structures using lamination parameters* (Doctoral dissertation). Delft University of Technology.

Martins, J. (2012). Sensitivity Analysis. http://aero-comlab.stanford.edu/jmartins/aa222/aa222sa.pdf

Miki, M. (1982). Material design of composite laminates with required in-plane elastic properties. *Progress in science and engineering of composites*, *2*, 1725–1731.

Mlejnek, H. P. (1992). Some aspects of the genesis of structures. *Structural Optimization*, *5*(1-2), 64–69. https://doi.org/10.1007/BF01744697

Nagy, A., Abdalla, M., & Gurdal, Z. (2010). *Design of anisotropic composite shells using an isogeometric approach*. https://doi.org/10.2514/6.2010-9181

Nagy, A., Ijsselmuiden, S. T., & Abdalla, M. (2013). Isogeometric design of anisotropic shells: Optimal form and material distribution. *Computer Methods in Applied Mechanics and Engineering*, *264*, 145–162. https://doi.org/10.1016/j.cma.2013.05.019

Novotny, A. A., Feijóo, R. A., Taroco, E., & Padra, C. (2003). Topological sensitivity analysis. *Computer Methods in Applied Mechanics and Engineering*, *192*(7), 803–829. https://doi.org/https://doi.org/10.1016/S0045-7825(02)00599-6

Peeters, D., Hong, Z., & Abdalla, M. (2018). A compliance approximation method applied to variable stiffness composite optimisation. *Structural and Multidisciplinary Optimization*, *58*(5), 1981–2001. https://doi.org/10.1007/s00158-018-2007-2

Peeters, D., van Baalen, D., & Abdallah, M. (2015). Combining topology and lamination parameter optimisation. *Structural and Multidisciplinary Optimization*, *52*(1), 105–120. https://doi.org/10.1007/s00158-014-1223-7

Peeters, D., Hesse, S., & Abdalla, M. (2015). Stacking sequence optimisation of variable stiffness laminates with manufacturing constraints. *Composite Structures*, *125*, 596–604. https://doi.org/10.1016/j.compstruct.2015.02.044

Raju, G., Wu, Z., & Weaver, P. M. (2014). On further developments of the feasible region of lamination parameters for composite laminates. *55th AIAA/ASMe/ASCE/AHS/SC Structures, Structural Dynamics, and Materials Conference*, (January). https://doi.org/10.2514/6.2014-1374

Reddy, J. N. (2004). *Mechanics of Laminated Composite Plates and Shells*. CRC Press. https://doi.org/https://doi.org/10.1201/b12409

Rozvany, G. (2001). Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. *Structural and Multidisciplinary Optimization*, *21*(2), 90–108. https://doi.org/10.1007/s001580050174

Rozvany, G. (2009). A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization*, *37*(3), 217–237. https://doi.org/10.1007/s00158-007-0217-0

Schittkowski, A. P. K. (2005). DFNLP : A Fortran Implementation of an SQP-Gauss-Newton Algorithm - User s Guide , Version 2 . 0 -, 1–32.

Setoodeh, S., Abdalla, M., & Gürdal, Z. (2006a). Design of variable-stiffness laminates using lamination parameters. *Composites Part B: Engineering*, *37*(4-5), 301–309. https://doi.org/10.1016/j.compositesb.2005.12.001

Setoodeh, S., Abdalla, M., & Gürdal, Z. (2006b). Approximate feasible regions for lamination parameters. *Collection of Technical Papers - 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, *2*(September), 814–822. https://doi.org/10.2514/6.2006-6973

Setoodeh, S., Blom, A., Abdalla, M., & Gürdal, Z. (2006). Generating Curvilinear Fiber Paths from Lamination Parameters Distribution. *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 14th AIAA/ASME/AHS Adaptive Structures Conference 7th*, *5*, 3440–3452. https://doi.org/10.2514/6.2006-1875

Sigmund, O. (1997). On the design of compliant mechanisms using topology optimization. *Mechanics of Structures and Machines*, *25*(4), 493–524. https://doi.org/10.1080/08905459708945415

Sigmund, O. (2001). A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, *21*(2), 120–127. https://doi.org/10.1007/s001580050176

Sigmund, O. (2007). Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, *33*(4-5), 401–424. https://doi.org/10.1007/s00158-006-0087-x

Sigmund, O., & Maute, K. (2013). Topology optimization approaches: A comparative review. *Structural and Multidisciplinary Optimization*, *48*(6), 1031–1055. https://doi.org/10.1007/s00158-013-0978-6

Sigmund, O., & Petersson, J. (1998). Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, *16*(1), 68–75. https://doi.org/10.1007/BF01214002

Sigmund, O. (2009). Manufacturing tolerant topology optimization. *Acta Mechanica Sinica/Lixue Xuebao*, *25*(2), 227–239. https://doi.org/10.1007/s10409-009-0240-z

Sokolowski, J., & Zochowski, A. (1999). On the Topological Derivative in Shape Optimization. *SIAM Journal on Control and Optimization*, *37*. https://doi.org/10.1137/S0363012997323230

Svanberg, K. (2002). A Class of Globally Convergent Optimization Methods Based on Conservative Convex Separable Approximations. *SIAM Journal on Optimization*, *12*, 555–573. https://doi.org/10.1137/S1052623499362822

Tsai, S., & Hahn, H. (1980). *Introduction to Composite Materials* (2nd). Routledge. https://doi.org/https://doi.org/10.1201/9780203750148

van Campen, J., Kassapoglou, C., & Gürdal, Z. (2012). Generating realistic laminate fiber angle distributions for optimal variable stiffness laminates. *Composites Part B: Engineering*, *43*(2), 354–360. https://doi.org/10.1016/j.compositesb.2011.10.014

van Tooren, M. J., & Elham, A. (2016). Optimization of variable stiffness composite plates with cut-outs subjected to compression, tension and shear using an adjoint formulation. *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, (January), 1–17. https://doi.org/10.2514/6.2016-1970

Wallin, M., Ristinmaa, M., & Askfelt, H. (2012). Optimal topologies derived from a phase-field method. *Structural and Multidisciplinary Optimization*, *45*(2), 171–183. https://doi.org/10.1007/s00158-011-0688-x

Wang, F., Lazarov, B., & Sigmund, O. (2011). On projection methods, convergence and robust formulations in topology optimization. *Structural and Multidisciplinary Optimization*, *43*(6), 767–784. https://doi.org/10.1007/s00158-010-0602-y

Wang, M., Wang, X., & Guo, D. (2003). A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, *192*(1-2), 227–246. https://doi.org/10.1016/S0045-7825(02)00559-5

Wu, Z., Raju, G., & Weaver, P. M. (2013). Feasible region of lamination parameters for optimization of variable angle tow (VAT) composite plates. *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 1–10. https://doi.org/10.2514/6.2013-1481

Wu, Z., Raju, G., & Weaver, P. M. (2015). Framework for the buckling optimization of variable-angle tow composite plates. *AIAA Journal*, *53*(12), 3788–3804. https://doi.org/10.2514/1.J054029

Xie, Y. M., & Steven, G. P. (1993). A simple evolutionary procedure for structural optimization. *Computers & Structures*, *49*(5), 885–896. https://doi.org/https://doi.org/10.1016/0045-7949(93)90035-C

Young, V., Querin, O., Steven, G., & Xie, Y. (1999). 3D and multiple load case bi-directional evolutionary structural optimization (BESO). *Structural Optimization*, *18*, 183–192. https://doi.org/10.1007/BF01195993

Zhou, M., & Rozvany, G. (1991). The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, *89*(1-3), 309–336. https://doi.org/10.1016/0045-7825(91)90046-9